POLITECNICO DI TORINO

Data Science and Engineering



Master's Degree

Time-Series Anomaly Detection

Supervisors

Prof. Paolo GARZA

Candidate

Sepehr ALEMZADEH

July 2026

Summary

Modern railway systems rely on dense sensor networks to continuously monitor critical subsystems during daily operation. Among these, the traction system plays a central role in motion generation and power transmission. Abnormal behavior within this subsystem can directly impact train performance, operational reliability, and passenger safety. Detecting anomalies in multivariate time-series data collected from traction system sensors is therefore of substantial practical importance. However, in real industrial environments, reliable anomaly labels are scarce, rendering supervised detection approaches impractical and motivating the adoption of unsupervised methodologies.

This thesis investigates unsupervised anomaly detection in high-resolution multivariate time-series data obtained from a railway traction system operated by CRRC in collaboration with BLUE Engineering. The dataset consists of 75 daily recordings from 88 sensors, sampled at one-second resolution, covering the period from February 2024 to June 2025. A structured preprocessing pipeline is developed, including strict temporal grid enforcement, systematic gap detection and classification, differentiated imputation strategies (linear interpolation for short-duration gaps and K-Nearest Neighbors imputation for long-duration gaps), and sensor-wise z-score normalization to ensure statistical comparability across channels.

Four deep learning models representing two distinct modeling paradigms are implemented and comparatively evaluated. The first paradigm comprises deterministic reconstruction based models, namely the LSTM Autoencoder and the Convolutional Autoencoder. The second paradigm includes probabilistic models, specifically the Variational Autoencoder and the Recurrent Deep State-Space Model. All models are assessed under identical preprocessing, windowing, scoring, and thresholding conditions within a unified experimental framework. Evaluation is conducted through anomaly score analysis, percentile-based thresholding, event-level aggregation, and a suite of unsupervised quality metrics, including score separability, event coherence, temporal stability, and threshold robustness.

The experimental results demonstrate that model architecture significantly influences anomaly detection behavior. Deterministic models exhibit smooth anomaly score trajectories proportional to reconstruction error, whereas probabilistic models capture deviations in distributional structure and temporal dynamics, producing sharper and more concentrated anomaly events. Cross-model agreement on highly anomalous operational periods further supports detection consistency. Sensitivity analysis across key hyperparameters confirms the robustness of the observed behavioral patterns.

Overall, the findings indicate that modeling assumptions—particularly those related to uncertainty representation and temporal dynamics—play a critical role in unsupervised anomaly detection for industrial time-series systems. The study provides practical guidance for model selection in real-world railway monitoring and predictive maintenance applications.

Acknowledgements

I would like to express my sincere gratitude to my university supervisor, **Professor Paolo Garza**, for his valuable guidance, support, and feedback throughout the development of this thesis.

I would also like to thank **BLUE Engineering**, the company where I carried out this project, for providing me with the opportunity to work on this research in a professional environment. My sincere thanks go to my company supervisor and manager, **Giuseppe Chirico**, for his support, guidance, and encouragement during the project.

Finally, I would like to thank all those who supported me directly or indirectly throughout this work.

Table of Contents

List of Tables x

List of Figures xi

Acronyms xiii

1 Introduction 1

1.1 Background and Motivation 1

1.2 Problem Context and Challenges 2

1.3 Research Objectives 3

1.4 Formal Problem Statement 3

1.4.1 Input Data 3

1.4.2 Assumptions 4

1.4.3 The Anomaly Detection Task 5

1.4.4 Expected Output 6

1.4.5 Success Criteria 6

1.5 Approach Overview 7

1.6 Contributions 7

1.7 Thesis Structure 8

2 Related Work 10

2.1 Foundations of Anomaly Detection 10

2.2 Industrial Multivariate Time-Series Anomaly Detection 11

2.3 Reconstruction-Based Deep Learning Methods 12

2.4 Limitations of Reconstruction-Based Models 13

2.5 Probabilistic Approaches to Anomaly Detection 13

2.6 Probabilistic State-Space Models 14

2.7 Graph and Attention-Based Models 15

2.8 Evaluation, Thresholding, and Benchmarking 15

2.9 Scope and Positioning of This Work 16

3 Dataset and Preprocessing 18

3.1 Preprocessing Pipeline Design 18

3.1.1 Enforcing a Strict Temporal Grid 18

3.1.2 Gap Detection and Classification 19

3.1.3 Imputation Strategies 19

3.1.4 Results 20

3.2 Normalization 21

3.3 Dataset Overview 22

3.4 Exploratory Data Analysis 22

3.4.1 Sensor Signal Distributions 23

3.4.2 Temporal Patterns and Daily Profiles 24

3.4.3 Sensor Correlations 25

3.4.4 Data Completeness and Recording Characteristics 26

3.5 Final Preprocessed Dataset 28

4 Anomaly Detection Models 29

4.1 Window-Based Input Formulation 29

4.2 LSTM Autoencoder (LSTM-AE) 32

4.2.1 Architecture 32

4.2.2 Training Objective 34

4.2.3 Anomaly Score 34

4.3 Convolutional Autoencoder (Conv-AE) 34

4.3.1 Architecture 34

4.3.2 Training Objective and Anomaly Score 36

4.4 Variational Autoencoder (VAE-Prob) 36

4.4.1 Motivation 36

4.4.2 Architecture 37

4.4.3 Training Objective 38

4.4.4 Anomaly Score 39

4.5 Recurrent Deep State-Space Model (RDSSM) 39

4.5.1 Motivation 39

4.5.2 Architecture 40

4.5.3 Training Objective 42

4.5.4 Anomaly Score 43

5 Experimental Setup 44

5.1 Experimental Pipeline Overview 44

5.2 Time-Series Windowing Strategy 45

5.3 Data Splitting Strategy 46

5.4 Training Protocol 47

5.5 Anomaly Score Computation 48
5.6 Thresholding Strategy 49
5.7 Event-Level Aggregation 49
5.8 Unsupervised Evaluation Metrics 50
5.8.1 Score Separability 50
5.8.2 Event-Level Metrics 50
5.8.3 Temporal Stability 51
5.8.4 Threshold Robustness 51
5.8.5 Percentage of Anomalous Time 51
5.9 Cross-Model Comparison Framework 51
5.10 Sensitivity Analysis Design 52
5.11 Synthetic Anomaly Benchmark Design 53
5.12 Improvement Design 54

6 Experimental Results 55
6.1 Overview of Evaluation Scope 55
6.2 Anomaly Score Behavior 56
6.3 Threshold Analysis 58
6.4 Maximum Anomaly Scores 58
6.5 Quantitative Model Comparison 59
6.6 Event-Level Results 61
6.7 Temporal Stability of Anomaly Scores 63
6.8 Sensitivity Analysis 64
6.8.1 Effect of Window Size 64
6.8.2 Effect of Threshold Percentile 64

6.8.3 Effect of Latent Dimension 65
6.9 Cross-Model Agreement Analysis 65
6.10 Synthetic Anomaly Benchmark 66
6.11 Improvement Results 69
6.12 Delta Comparison Improved vs Baseline 70
6.13 Summary of Key Findings 71

7 Discussion 73

7.1 Deterministic vs. Probabilistic Modeling Paradigms 74
7.2 Interpretation of Numerical Scale Differences 74
7.3 Temporal Sensitivity and Pattern Capture 75
7.4 Event Fragmentation vs. Coherence 75
7.5 Cross-Model Agreement and Confidence 76
7.6 Stability Under Noise and Data Quality 76
7.7 Strengths and Limitations of Each Model 76
7.8 Limitations of the Study 77
7.9 Practical Implications 77
7.10 Improvement Study 78

8 Conclusion and Future Work 79

8.1 Summary of the Work 79
8.2 Conclusions Drawn from the Results 80
8.3 Contributions 82
8.4 Limitations 82
8.5 Future Work 83

Bibliography 85

List of Tables

3.1 Summary of dataset characteristics 22

3.2 Sensor statistics 24

4.1 Model architecture comparison summary 43

5.1 Representative split summary 47

5.2 Training hyperparameters 48

5.3 Sensitivity analysis configurations 52

6.1 Threshold Analysis 58

6.2 Maximum Anomaly score 59

6.3 Aggregated Metrics 59

6.4 Event-level statistics on 2024-12-05 63

6.5 Jaccard similarity 65

6.6 Synthetic anomaly benchmark summary 67

6.7 Improved synthetic anomaly benchmark summary 70

6.8 Calibrated post-processing parameters 70

6.9 Delta comparison between improved and baseline synthetic results . 70

7.1 Strengths and limitations of each model 76

List of Figures

3.1 Long Gap imputation sample 1 21
3.2 Long Gap imputation sample 2 21
3.3 sensor distribution across 75-file dataset 23
3.4 daily sensor profiles on 2024-12-05 24
3.5 cross-month comparison for sensor ZT31171 25
3.6 correlation heatmap of 88 sensors 26
3.7 Daily recording duration across 75-day dataset 27
3.8 Distribution of daily recording durations 27
3.9 Day signals before and after Normalization 28

4.1 Windowing 31
4.2 T time steps 31
4.3 LSTM-AE Architecture 33
4.4 Convolutional-AE Architecture 36
4.5 Variational-AE Architecture 38
4.6 RDSSM Architecture 42

5.1 Experimental pipeline overview 45

6.1 Anomaly score behavior on the representative day 56

6.2 Anomaly score distributions with their corresponding anomaly thresholds. 57
6.3 Boxplot comparison of event-level anomaly detection 60
6.4 normalized cross-model metric comparison 61
6.5 Event timeline on Final-2024-12-05 62
6.6 Threshold percentile sensitivity 64
6.7 Cross-model agreement(jaccard) 66
6.8 Synthetic anomaly benchmark: F1 by anomaly family 68
6.9 Synthetic anomaly benchmark: event detection rate 69
6.10 Synthetic anomaly benchmark: AUROC 69

Acronyms

AI
artificial intelligence

ML
machine learning

DL
deep learning

IoT
Internet of Things

AE
autoencoder

LSTM
Long Short-Term Memory

LSTM-AE
Long Short-Term Memory Autoencoder

Conv-AE
Convolutional Autoencoder

VAE
Variational Autoencoder

VAE-Prob

Probabilistic Variational Autoencoder

RDSSM

Recurrent Deep State-Space Model

RNN

recurrent neural network

GRU

Gated Recurrent Unit

CNN

convolutional neural network

MLP

multilayer perceptron

MSE

mean squared error

MAE

mean absolute error

RMSE

root mean squared error

ELBO

Evidence Lower Bound

KL

Kullback–Leibler

KNN

K-Nearest Neighbors

AUROC

Area Under the Receiver Operating Characteristic Curve

AUPRC
: Area Under the Precision–Recall Curve

ROC
: Receiver Operating Characteristic

PR
: precision–recall

F1
: F1 score

TP
: true positive

FP
: false positive

TN
: true negative

FN
: false negative

EDA
: exploratory data analysis

PDF
: Portable Document Format

SVG
: Scalable Vector Graphics

CRRC
: China Railway Rolling Stock Corporation

BLUE
: BLUE Engineering

Chapter 1

Introduction

1.1 Background and Motivation

Modern railway systems are increasingly equipped with dense networks of Internet of Things (IoT) sensors that continuously monitor subsystem health during daily operations. These sensors generate large volumes of high-resolution multivariate time-series data, recording physical quantities such as temperature, pressure, voltage, current, and vibration signals. Systematic analysis of these data streams enables the detection of abnormal patterns that may indicate emerging faults, component degradation, or irregular operating conditions. these kind of early detection capabilities are essential for enhancing operational reliability, improving maintenance planning, and ensuring passenger safety.

Among railway subsystems, the traction system plays a particularly important role. it converts electrical energy into the mechanical power needed to move the train and delivers that power to the wheels, making it the key component responsible for train propulsion. Malfunctions within the traction system—whether due to mechanical wear, electrical failures, or environmental influences—can significantly affect performance and, in severe cases, compromise safety. Consequently, automated anomaly detection in traction system sensor data represents a critical component of modern predictive maintenance strategies.

Although industrial systems often generate large volumes of sensor data, detecting anomalies in real-world environments remains challenging. unlike controlled laboratory settings or benchmark datasets, real operational data is often noisy, inconsistent, and influenced by changing conditions. these practical challenges shaped the methodology and experimental framework presented in this thesis

1.2 Problem Context and Challenges

The dataset analyzed in this work consists of multivariate time-series data from 88 sensors installed on the traction systems of trains operated by CRRC, collected in collaboration with BLUE Engineering. The data span approximately fourteen months, from February 2024 to June 2025, and include 75 daily recordings selected across the observation period.

Three principal challenges characterize this problem setting.

Data quality verification and preprocessing. Although sensors are configured to record measurements at a frequency of one sample per second (1 Hz), temporal completeness and consistency cannot be assumed without verification. A systematic preprocessing pipeline is therefore required to enforce a strict temporal grid, detect potential gaps or irregularities, and normalize heterogeneous sensor scales. While the current dataset was verified to be temporally complete at 1 Hz with no missing values, the pipeline is deliberately designed to handle potential irregularities in future deployments.

Absence of reliable anomaly labels. Only a limited number of maintenance events and operational incidents have been manually documented. These records are sparse, incomplete, and lack the granularity required for supervised model training or rigorous quantitative evaluation. As a result, the problem must be approached in a fully unsupervised setting, where models learn patterns of normal system behavior from predominantly healthy data and identify deviations without explicit anomaly labels.

Complex temporal dynamics and high dimensionality. The sensor data exhibit variability across multiple temporal scales. Certain anomalies manifest abruptly, such as sudden spikes or rapid state transitions, whereas others evolve gradually over extended time horizons, reflecting progressive degradation processes. Effective anomaly detection requires models capable of capturing both short-term fluctuations and long-term temporal dependencies, as well as cross-sensor interactions within high-dimensional multivariate signals.

1.3 Research Objectives

The overarching goal of this thesis is to investigate unsupervised deep learning methods for anomaly detection in industrial multivariate time-series data. The specific objectives are as follows:

1. **To design and implement a robust preprocessing pipeline** that verifies temporal consistency, addresses potential missing data, and standardizes sensor scales to enable reliable downstream modeling.
2. **To implement and comparatively evaluate multiple deep learning architectures** for unsupervised anomaly detection, spanning deterministic reconstruction-based models and probabilistic generative models.
3. **To establish a structured evaluation framework** based on anomaly score analysis, event-level aggregation, temporal stability assessment, and unsupervised quality metrics that do not rely on ground-truth labels.
4. **To analyze the impact of modeling assumptions**—particularly regarding uncertainty representation, latent variable modeling, and temporal dependency structures—on anomaly detection behavior in industrial settings.

1.4 Formal Problem Statement

The previous section described the practical context of the problem. This section defines the anomaly detection task in a precise, formal manner, specifying the input data, the underlying assumptions, and the expected output of the proposed approach. Establishing this formalization early serves two purposes: it removes ambiguity about what the models are asked to do, and it provides the common notation used throughout the remainder of the thesis.

1.4.1 Input Data

The input to the system is a collection of multivariate time series recorded by the sensors of the traction system. At each time instant t, the state of the system is described by a vector of sensor readings:

$$x_t = (x_{t,1}, x_{t,2}, \ldots, x_{t,D}) \in \mathbb{R}^D, \qquad D = 88,$$

where each component $x_{t,d}$ is the value measured by sensor d at time t. The sensors measure heterogeneous physical quantities, such as temperatures, currents, and voltages, so the components of x_t have different units and value ranges. The data are sampled at a nominal frequency of 1 Hz, meaning that consecutive observations are one second apart.

The full dataset consists of $J = 75$ daily recordings, collected between February 2024 and June 2025:

$$\mathcal{D} = \{X^{(1)}, X^{(2)}, \ldots, X^{(J)}\}, \qquad X^{(j)} = (x_1^{(j)}, x_2^{(j)}, \ldots, x_{T_j}^{(j)}),$$

where $X^{(j)}$ is the j-th daily recording and T_j is its length in seconds. Recording durations vary across days, from short sessions to full 24-hour recordings.

Because the models operate on fixed-length inputs, each daily series is segmented into overlapping windows of length W, with $W = 120$ s in this work, using a sliding stride of one second:

$$X_t = [x_{t-W+1}, x_{t-W+2}, \ldots, x_t] \in \mathbb{R}^{W \times D}.$$

A window is therefore the fundamental unit of analysis: it represents what the system looked like during the last two minutes.

1.4.2 Assumptions

The formulation of the task relies on the following assumptions, which reflect the industrial conditions described in Section 1.2. and are consistent with common unsupervised and one-class anomaly detection settings. [1]

Assumption 1: Predominantly normal data. Most of the recorded data represents normal system operation, while anomalous events occur occasionally. This makes unsupervised learning a practical choice, as the models trained mainly on normal operating data and therefore learns the typical behavior of the system. [2] Any observations that deviate significantly from this learned pattern can then be identified as potential anomalies

Assumption 2: No reliable labels. No trustworthy, fine-grained annotations are available indicating which time instants are anomalous. Consequently, the models cannot be trained or directly validated with supervised objectives, and evaluation must rely on unsupervised quality metrics and, complementarily, on controlled synthetic anomalies injected only at evaluation time.

Assumption 3: Temporal regularity. After preprocessing, each daily recording lies on a strict one-second temporal grid with no missing values, so that every window X_t is complete and well defined. This follows the common formulation of anomaly detection in time-series data, where observations are analyzed as ordered temporal sequences [3].

Assumption 4: Statistical comparability of sensors. After sensor-wise z-score normalization, all sensor channels operate on comparable numerical scales, so that no single sensor dominates the learning objective purely because of its physical unit or magnitude.

Assumption 5: Stationarity of normal behavior at the window scale. Normal operating patterns observed during training are representative of normal behavior at evaluation time. Slow, long-term evolution of the system, such as seasonal effects, is possible, but within the two-minute window scale, the statistical structure of normal behavior is assumed to be stable enough to be learned.

1.4.3 The Anomaly Detection Task

Informally, an anomaly is a time period in which the system behaves in a way that is inconsistent with its learned normal behavior. Formally, let $p_{\text{normal}}(X)$ denote the unknown probability distribution of windows generated by the system under normal operating conditions. An anomalous window is a window that is unlikely under this distribution, i.e., a window lying in a low-probability region of p_{normal}.

Since p_{normal} is unknown, the task is to learn, from the data alone, a scoring function

$$f_\theta : \mathbb{R}^{W \times D} \to \mathbb{R}, \qquad s_t = f_\theta(X_t),$$

parameterized by a deep neural network with parameters θ, such that the anomaly score s_t is low for windows consistent with normal behavior and high for windows that deviate from it.

The four models studied in this thesis define f_θ in different ways. The deterministic models define s_t as a reconstruction error, while the probabilistic models define s_t through a likelihood-based criterion. In both cases, the score can be read as a measure of how surprising the current window is with respect to what the model has learned to consider normal.

1.4.4 Expected Output

The proposed approach produces three levels of output, each obtained from the previous one.

1. **Continuous anomaly scores.** For every time step t of every daily recording, a scalar score $s_t \in \mathbb{R}$ quantifies the degree of abnormality of the window ending at t. Scores are smoothed with a 30-second rolling average to suppress high-frequency noise.

2. **Binary anomaly decisions.** A threshold τ, computed as the 99.5th percentile of the training-score distribution, converts the continuous score into a binary flag:
$$a_t = \begin{cases} 1, & \text{if } s_t > \tau \quad \text{(anomalous)}, \\ 0, & \text{otherwise} \quad \text{(normal)}. \end{cases}$$

3. **Anomaly events.** Consecutive anomalous time steps, allowing gaps of at most 10 seconds, are merged into contiguous events
$$E_k = \{t_{\text{start}}^{(k)}, \ldots, t_{\text{end}}^{(k)}\},$$
each characterized by a start time, an end time, and a duration. Events are the operationally meaningful output: rather than thousands of individual flagged seconds, the maintenance engineer receives a short list of time intervals during which the system behaved abnormally, ranked by their peak anomaly score.

1.4.5 Success Criteria

Given the absence of ground-truth labels, a solution to this task cannot be judged by classification accuracy alone. Instead, a good solution is one that:

1. assigns clearly separated scores to normal and abnormal behavior, i.e., score separability;

2. produces temporally coherent events rather than fragmented isolated flags;

3. remains stable under reasonable variations of the detection threshold;

4. detects controlled synthetic anomalies of known location and type with high precision and recall and with a short detection delay.

These criteria are formalized as evaluation metrics in the experimental evaluation chapter.

1.5 Approach Overview

This work follows a structured experimental pipeline.

First, a comprehensive preprocessing stage enforces temporal regularity, reconstructs the timeline where necessary, performs systematic gap detection, and applies differentiated imputation strategies. Exploratory data analysis is conducted to characterize statistical properties, correlations, and temporal patterns within the sensor signals.

Subsequently, four deep learning models are implemented and trained in a fully unsupervised manner:

- **LSTM Autoencoder (LSTM-AE):** A recurrent reconstruction-based architecture designed to capture long-range temporal dependencies.
- **Convolutional Autoencoder (Conv-AE):** A convolutional reconstruction-based model focusing on local temporal patterns and feature extraction.
- **Variational Autoencoder (VAE-Prob):** A probabilistic generative model with a stochastic latent space that enables uncertainty-aware anomaly scoring.
- **Recurrent Deep State-Space Model (RDSSM):** A probabilistic state-space model that explicitly represents latent temporal dynamics and stochastic state transitions.

All models are evaluated under identical experimental conditions—including windowing strategy, training protocol, anomaly scoring, threshold selection, and event aggregation—to ensure that observed performance differences are attributable solely to architectural and probabilistic design choices.

1.6 Contributions

The primary contributions of this thesis are summarized as follows:

1. The development of a **robust preprocessing pipeline** tailored to high-resolution industrial sensor data, incorporating structured temporal verification and differentiated imputation strategies, including linear interpolation for short gaps and K-Nearest Neighbors reconstruction for long gaps.

2. The establishment of a **unified experimental framework** enabling a fair and controlled comparison between deterministic and probabilistic anomaly detection models.

3. A **comprehensive behavioral analysis** of four deep learning model families using unsupervised evaluation metrics, including score separability, event coherence, temporal stability, and threshold robustness.

4. The derivation of **practical insights** into how architectural design choices—particularly those concerning temporal modeling, uncertainty representation, and latent dynamics—shape anomaly detection behavior in real-world industrial time-series systems.

1.7 Thesis Structure

The remainder of this thesis is organized as follows:

- **Chapter 2** reviews related work in time-series anomaly detection, covering both deterministic and probabilistic approaches and positioning this study within the broader literature.

- **Chapter 3** presents the dataset description, exploratory data analysis, and the detailed preprocessing pipeline, including gap detection, imputation strategies, and normalization.

- **Chapter 4** introduces the four anomaly detection models, detailing their architectural design, mathematical formulations, and anomaly scoring mechanisms.

- **Chapter 5** describes the experimental setup, including windowing strategy, training procedures, evaluation methodology, and unsupervised quality metrics.

- **Chapter 6** reports the experimental results, including quantitative comparisons, anomaly score analysis, sensitivity studies, and cross-model agreement assessment.

- **Chapter 7** discusses the findings, analyzes trade-offs between modeling paradigms, and outlines practical implications.
- **Chapter 8** concludes the thesis and proposes directions for future research.

Chapter 2

Related Work

This chapter reviews prior work on anomaly detection for multivariate time-series data, with emphasis on industrial sensor monitoring and unsupervised deep learning methods. The literature is organized from general anomaly detection foundations to reconstruction-based neural models, probabilistic approaches, state-space models, graph and attention-based models, and evaluation strategies. This organization reflects the central aim of this thesis: to compare deterministic and probabilistic anomaly detection models under a unified experimental framework for railway traction-system sensor data.

2.1 Foundations of Anomaly Detection

Anomaly detection concerns the identification of observations, patterns, or temporal segments that deviate from expected system behavior. In industrial monitoring, such deviations may indicate sensor malfunction, abnormal operating conditions, component degradation, or early signs of system failure. Chandola et al. [1] provide one of the most widely cited surveys of anomaly detection, classifying methods into statistical, distance-based, density-based, clustering-based, and model-based approaches. Their taxonomy is useful for this thesis because modern deep learning methods can be interpreted as model-based approaches that learn a representation of normality from historical data.

Novelty detection is especially relevant when abnormal examples are rare or unavailable. Pimentel et al. [4] define novelty detection as the task of identifying test data that differ from the training data, typically under the assumption that

training data mostly represent normal behavior. This setting matches many real industrial applications, where collecting large volumes of normal sensor data is feasible, but obtaining reliable labels for failures is difficult. In railway traction systems, anomalies are sparse, maintenance logs may be incomplete, and fault labels may not be aligned with sensor timestamps. These conditions motivate the use of unsupervised or semi-supervised anomaly detection methods.

More recent work has focused specifically on time-series anomaly detection. Blazquez-Garcia et al. [3] review outlier and anomaly detection methods for time-series data and emphasize that temporal order introduces challenges not present in static data. An observation may be normal in isolation but anomalous in context, depending on preceding and following values. This distinction is central to multivariate railway sensor data, where abnormal behavior may appear as a temporal pattern, a cross-sensor inconsistency, or a change in operating regime rather than a single extreme value.

2.2 Industrial Multivariate Time-Series Anomaly Detection

Industrial systems increasingly rely on dense sensor networks to monitor physical processes continuously. These data streams are usually multivariate, high-dimensional, noisy, and non-stationary. They also contain dependencies across sensors and across time.[3] As a result, anomaly detection in industrial time series requires methods capable of modeling temporal dynamics, cross-variable relationships, and changes in operating conditions.[5]

Deep learning has become prominent in this area because neural models can learn nonlinear representations from high-dimensional sensor streams. Recent surveys emphasize that deep anomaly detection for multivariate time series commonly relies on reconstruction, prediction, representation learning, and probabilistic modeling. They also highlight persistent challenges such as label scarcity, evaluation inconsistency, model interpretability, and robustness under changing operating conditions. These challenges are directly reflected in this thesis, where the available railway data are real industrial recordings without dense ground-truth anomaly labels.[6]

Although this thesis is not focused on online deployment, practical monitoring issues still appear in offline evaluation: thresholds must be calibrated, anomaly scores must be interpreted carefully, and model behavior must be assessed beyond raw reconstruction error. This is particularly important in railway traction-system data,

where candidate anomalies may appear as gradual drifts, short local disturbances, or changes in the relationship between multiple sensor channels.

2.3 Reconstruction-Based Deep Learning Methods

A major family of deep anomaly detection methods is based on reconstruction. Autoencoders learn to compress input data into a latent representation and reconstruct the original input from that compressed form. When trained mainly on normal data, they are expected to reconstruct normal patterns accurately and produce larger reconstruction errors for anomalous inputs. Sakurada and Yairi [7] demonstrated the use of autoencoders for anomaly detection with nonlinear dimensionality reduction, comparing them with linear approaches such as principal component analysis. This work is important because it established autoencoders as practical tools for detecting subtle anomalies in sensor-like data.

For sequential data, recurrent neural networks extend the reconstruction idea by explicitly modeling temporal dependencies. Malhotra et al. [8] proposed an LSTM encoder–decoder framework for multi-sensor anomaly detection, where the model learns normal temporal behavior and detects anomalies through reconstruction error. This is closely related to the LSTM-AE model used in this thesis, which is designed to capture long-range temporal dependencies in railway traction-system signals.

Convolutional approaches provide another reconstruction-based direction. Munir et al. [9] introduced DeepAnT, a convolutional neural network approach for unsupervised time-series anomaly detection. Convolutional methods are attractive because they can efficiently capture local temporal patterns and abrupt deviations. This motivates the Conv-AE model used in this thesis, which focuses on local temporal structures and short-range signal distortions.

Beyond simple autoencoder and convolutional formulations, several works attempt to improve reconstruction-based anomaly detection by modeling richer temporal or cross-variable structure. MSCRED represents multivariate time series through multiscale signature matrices, allowing the model to capture both temporal patterns and correlations among variables [10]. USAD combines two autoencoders with adversarial training to improve stability and sensitivity in multivariate time-series monitoring [11]. These approaches are relevant to this thesis because they show that reconstruction error remains useful, but its effectiveness depends strongly on the architecture used to encode temporal and inter-sensor dependencies.

2.4 Limitations of Reconstruction-Based Models

Despite their widespread use, deterministic reconstruction-based approaches exhibit several limitations when applied to complex industrial time-series data. First, reconstruction error may fail to capture subtle distributional anomalies that do not manifest as large amplitude deviations. For example, changes in correlation structure, timing, or latent operating mode may not significantly increase reconstruction loss, leading to missed detections [12]. This limitation is especially relevant in high-dimensional systems, where abnormality may be expressed through relationships among variables rather than through isolated sensor values.

Second, reconstruction-based models can be sensitive to noise, missing values, or imputed data segments. In such cases, reconstruction quality may deteriorate for reasons unrelated to true system anomalies, producing unstable anomaly scores. This issue matters in real industrial data because preprocessing choices, sensor noise, and temporal gaps can affect the anomaly score even when the underlying system is not faulty.

Third, threshold selection for reconstruction error is often heuristic and depends on the numerical scale of the model output. This complicates cross-model comparison and reduces interpretability. A threshold that is meaningful for one model may not be meaningful for another, even when both are trained on the same data. Finally, deterministic models lack explicit uncertainty modeling and therefore cannot distinguish between deviations caused by stochastic variability and those indicative of meaningful system abnormalities. These limitations motivate probabilistic frameworks that explicitly model uncertainty and data distributions.

2.5 Probabilistic Approaches to Anomaly Detection

Probabilistic deep learning methods extend deterministic models by learning distributions over latent representations rather than fixed point estimates. This allows anomaly detection to be based on likelihood estimation, uncertainty quantification, or divergence from learned probabilistic structures. Kingma and Welling [13] introduced the Variational Autoencoder (VAE), which models the latent representation as a probability distribution and trains by maximizing the Evidence Lower Bound (ELBO). In anomaly detection contexts, observations with low likelihood under the learned normal distribution, or with high divergence in latent space, can be flagged as anomalous.

The VAE family is relevant for anomaly detection because it provides a probabilistic criterion rather than only a reconstruction criterion, allowing anomalies to be interpreted as low-likelihood or poorly regularized latent observations. Researchers demonstrated the effectiveness of VAE-based anomaly detection in multivariate sensor settings, showing that likelihood-based scoring can identify anomalies that reconstruction error alone might miss.A VAE-based method was proposed for seasonal KPI anomaly detection, emphasizing the role of probabilistic reconstruction and latent regularization in noisy time-series environments.

Density-based deep models further connect representation learning with explicit anomaly scoring. DAGMM integrates an autoencoder with Gaussian mixture modeling, using both the learned low-dimensional representation and reconstruction features for density estimation [12]. This is important because it addresses a weakness of plain autoencoders: reconstruction error alone may not fully describe whether a point is statistically abnormal.

OmniAnomaly extends probabilistic anomaly detection to multivariate time series by combining recurrent neural networks, stochastic latent variables, and normalizing flows [14]. Its use of reconstruction probability rather than simple reconstruction error is particularly relevant to this thesis, because the VAE-Prob and RDSSM models similarly treat anomalies as observations that are unlikely under a learned probabilistic representation of normal behavior.

2.6 Probabilistic State-Space Models

A distinct class of probabilistic models approaches anomaly detection through state-space modeling. In these frameworks, observed sensor measurements are assumed to be generated by latent states that evolve over time according to stochastic transition dynamics. Anomalies are detected when observations are unlikely under the inferred latent state or when state evolution deviates from expected trajectories.

Krishnan et al. [15] introduced Deep Kalman Filters, which combine neural network parameterization with classical state-space modeling to capture nonlinear dynamics while maintaining probabilistic structure. This work laid the foundation for deep state-space models capable of modeling complex temporal dependencies with uncertainty quantification. These ideas are connected to variational recurrent neural networks, which combine latent-variable modeling with recurrent sequence processing and provide a foundation for probabilistic temporal representation learning [16].

Recurrent Deep State-Space Models (RDSSMs) extend this concept by incorporating

recurrent neural networks, such as GRU cells, to parameterize both latent state transitions and inference networks. These models provide a principled probabilistic interpretation while retaining the expressive power of deep recurrent architectures. For railway traction systems, this modeling perspective is particularly relevant because abnormal behavior may correspond to hidden regime changes or unusual state transitions rather than isolated sensor spikes.

2.7 Graph and Attention-Based Models

Recent work has increasingly focused on explicitly modeling relationships among variables in multivariate time series. This is important in sensor networks because an anomaly may not appear as an extreme value in one signal, but as an unusual relationship between several signals. MTAD-GAT combines temporal modeling with graph attention to learn dependencies across both time and features [17]. Similarly, GDN learns graph-based sensor relationships and uses deviations from expected inter-sensor behavior for anomaly detection [18]. These methods are especially relevant to industrial systems, where physical connections between components often create structured dependencies among sensor channels.

Transformer-based methods represent another recent direction. Anomaly Transformer introduces an association-discrepancy mechanism to distinguish normal and abnormal temporal associations [19]. TranAD uses Transformer encoders with self-conditioning and adversarial training for multivariate time-series anomaly detection [20]. Although this thesis does not implement graph or Transformer models, these works help position the selected architectures within the broader literature. They also motivate future extensions in which railway sensor relationships could be modeled explicitly through attention or graph structure.

2.8 Evaluation, Thresholding, and Benchmarking

Evaluation methodology is a central issue in time-series anomaly detection. In real industrial settings, dense and reliable labels are often unavailable, and anomalies usually occur as temporal events rather than isolated point labels.[21]. As a result, point-wise precision and recall can provide an incomplete picture of model behavior, since a detector may identify the correct anomalous episode but disagree with the exact annotated start or end time, leading to misleadingly poor point-level metrics. This limitation has motivated range- and event-aware evaluation measures that explicitly account for temporal anomaly intervals rather than treating each

timestamp independently [6].

Thresholding is another important challenge. Reconstruction errors, likelihood scores, and state-space anomaly scores may operate on very different numerical scales. Therefore, raw anomaly scores are not directly comparable across models.[6] This is especially relevant in this thesis, where deterministic reconstruction-based models and probabilistic models produce anomaly scores with different interpretations and ranges. A fair comparison therefore requires model-relative thresholding, sensitivity analysis, and event-level aggregation. Prior work has similarly emphasized that thresholding is part of the operational anomaly-detection pipeline and that benchmark conclusions can depend strongly on metric and threshold choices [5].

For this reason, this thesis evaluates models not only through raw anomaly scores, but also through threshold sensitivity, event aggregation, cross-model agreement, and controlled synthetic anomalies. This evaluation design reflects the practical reality of industrial anomaly detection: useful systems must not only produce high anomaly scores, but must also produce interpretable and operationally meaningful events. This caution is particularly important given evidence that commonly used time-series anomaly-detection benchmarks and labels may contain flaws that distort apparent progress and model rankings [22].

2.9 Scope and Positioning of This Work

Building upon the reviewed literature, this thesis conducts a systematic and controlled comparison of deterministic and probabilistic unsupervised anomaly detection models applied to real-world railway traction-system data. Reconstruction-based autoencoders, namely LSTM-AE and Conv-AE, serve as deterministic baselines. Probabilistic models, namely VAE-Prop and RDSSM, are evaluated to examine the influence of uncertainty modeling and explicit latent dynamics on anomaly detection behavior.[5]

Unlike many prior studies that rely on curated benchmark datasets with labeled anomalies, this work operates in a fully unsupervised industrial setting with real sensor variability, imputation artifacts, and no reliable ground-truth annotations.[6] The primary contribution lies not in proposing a new architecture, but in establishing a unified experimental framework that enables fair comparison across modeling paradigms under identical preprocessing, windowing, scoring, and thresholding conditions. This positioning follows recent calls for careful and reproducible evaluation protocols in time-series anomaly detection, where dataset properties, anomaly

definitions, and metric choices can strongly affect conclusions [22].

This positioning connects the thesis to several strands of prior work. It follows reconstruction-based anomaly detection through autoencoders, LSTM encoder-decoder models, and convolutional approaches; it incorporates probabilistic reasoning through VAE-style and state-space models; and it acknowledges recent graph and Transformer models as important extensions. The resulting comparison is intended to clarify how architectural assumptions shape anomaly detection behavior in high-resolution industrial railway sensor data.

Chapter 3

Dataset and Preprocessing

This chapter describes the dataset used in this study, presents an exploratory analysis of its statistical and temporal characteristics, and details the preprocessing pipeline developed to convert raw sensor recordings into clean, continuous time series suitable for unsupervised anomaly detection.

3.1 Preprocessing Pipeline Design

Although the dataset was verified to be temporally complete with no missing sensor values, a robust preprocessing pipeline was designed and applied to ensure reproducibility and to handle potential data irregularities that may occur in future collections. The pipeline consists of four main stages: (1) temporal grid enforcement, (2) gap detection and classification, (3) conditional imputation, and (4) normalization.

3.1.1 Enforcing a Strict Temporal Grid

Each daily dataset is reindexed onto a strict one-second grid to ensure exactly one row exists per second during the recording interval. Any missing values at grid timestamps are explicitly represented as *NaN*.

- Raw timestamps: $\{t_{i_1}, t_{i_2}, \ldots, t_{i_N}\}$
- Reindexed grid: $\{t_0, t_1, \ldots, t_M\}$

- Grid definition: $t_j = t_{\text{start}} + j \cdot \Delta t$
- Sampling interval: $\Delta t = 1$ s

For the present dataset, this verification confirmed alignment to the one-second grid across all 75 daily recordings.

3.1.2 Gap Detection and Classification

After grid enforcement, consecutive *NaN* values are grouped into contiguous gap segments. For each gap segment g_k, the following properties are recorded:

1. Start time: $t_{\text{start}}^{(k)}$
2. End time: $t_{\text{end}}^{(k)}$
3. Gap length: L_k
4. $L_k = t_{\text{end}}^{(k)} - t_{\text{start}}^{(k)} + 1$

Gaps are categorized by duration:

- Short gaps: $L_k < 60$
- Long gaps: $L_k \geq 60$

This threshold reflects a practical distinction: short gaps are often brief communication delays with limited information loss, while long gaps may indicate downtime or sensor faults requiring different treatment. In the present dataset, no gaps were detected, confirming high acquisition reliability.

3.1.3 Imputation Strategies

The pipeline includes conditional imputation methods that are applied only if missing segments are detected.

Short-gap imputation (linear interpolation).

Short gaps are filled using linear interpolation to preserve continuity and local trends.

$$\hat{x}_i = x_a + \frac{i}{n+1}(x_b - x_a), \qquad i = 1, 2, \ldots, n \tag{3.1}$$

where x_a is the last observed value before the gap, x_b is the first observed value after the gap, and n is the number of missing points.

Long-gap imputation (KNN-based).

Long gaps are filled using K-Nearest Neighbors imputation based on similarity to historical contexts. For each missing timestamp, a context vector is compared to historical context vectors, and the nearest neighbors are used for estimation.

Distance (Euclidean):

$$d(c_{\text{missing}}, c_i) = \sqrt{\sum_{d=1}^{D} \left(c_{\text{missing},d} - c_{i,d}\right)^2} \tag{3.2}$$

Value estimation:

$$\hat{x}_{\text{missing}} = \frac{1}{K} \sum_{j=1}^{K} x_j \tag{3.3}$$

To avoid introducing artificial cross-sensor dependencies that could bias anomaly detection, sensors are imputed independently rather than using cross-imputation across channels. Since no gaps were present, these imputation steps were not invoked for the current dataset, but remain integral for robustness.

3.1.4 Results

After applying the complete preprocessing pipeline, all datasets are transformed into fully continuous time series with a strict one-second resolution and no missing values. The absence of *NaN* values in the final dataset therefore reflects the outcome of the imputation process rather than the condition of the original raw data.

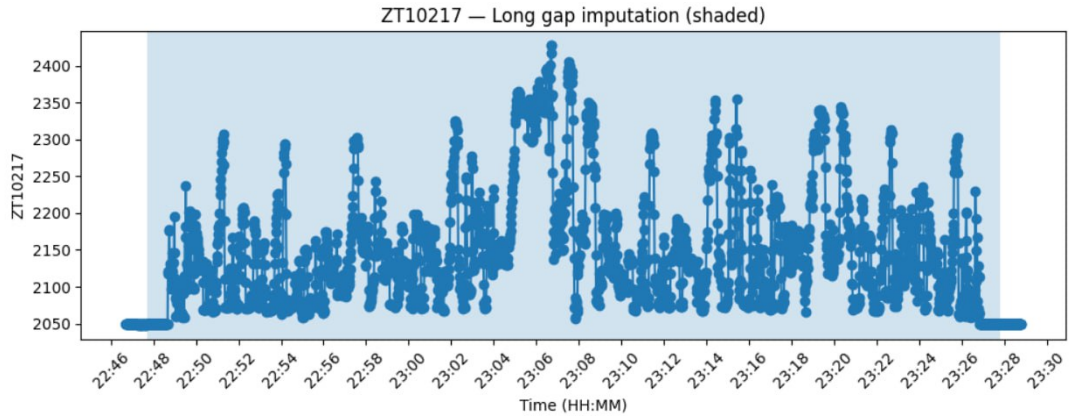


Figure 3.1: Long Gap imputation sample 1

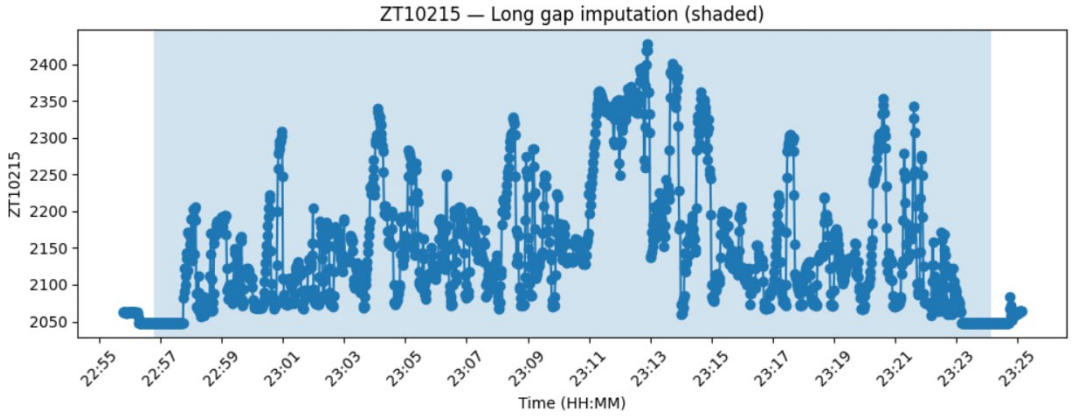


Figure 3.2: Long Gap imputation sample 2

3.2 Normalization

Normalization is critical because the 88 sensors measure different physical quantities and have widely varying units and value ranges. Without normalization, sensors with large numerical magnitude can dominate learning dynamics and anomaly scores.

Sensor-wise z-score normalization is applied using training-set statistics.

$$\tilde{x}_{t,d} = \frac{x_{t,d} - \mu_d}{\sigma_d} \tag{3.4}$$

where $x_{t,d}$ is sensor d at time t, and μ_d and σ_d are the mean and standard deviation of sensor d computed on the training data. The same (μ_d, σ_d) are applied consistently across all days to ensure comparability of normalized values and stability of the scoring pipeline.

3.3 Dataset Overview

After preprocessing and normalization, the dataset consists of clean multivariate time-series recordings collected from the traction system of a railway vehicle.

The data span from February 2024 to June 2025 and include 75 daily recordings. Each dataset is sampled at a uniform frequency of one observation per second and contains 88 sensor channels representing various physical and operational measurements.

Recording durations vary across days, ranging from short sessions to full 24-hour recordings, with an average duration of approximately 15.5 hours.

Table 3.1: Summary of dataset characteristics

Property	**Value**
Observation period	Feb 2024–June 2025
Number of daily datasets	75
Number of sensors	88
Nominal sampling rate	1 Hz
Typical daily recording duration	∼17.2 hours
Total approximate data points	∼408 million
Timestamp gaps	0.00%
Sensor NaN rate	0.00%

3.4 Exploratory Data Analysis

An exploratory analysis was performed to understand the statistical properties, temporal structure, and data quality of the sensor recordings. This analysis informs preprocessing choices and establishes baseline expectations for downstream anomaly detection behavior.

3.4.1 Sensor Signal Distributions

The 88 sensor channels exhibit heterogeneous statistical behavior, reflecting diverse physical quantities and operating modes. Temperature sensors typically display unimodal distributions concentrated around operational ranges. In contrast, current and voltage sensors often show multimodal distributions corresponding to discrete operating regimes such as acceleration, cruising, braking, and idle.

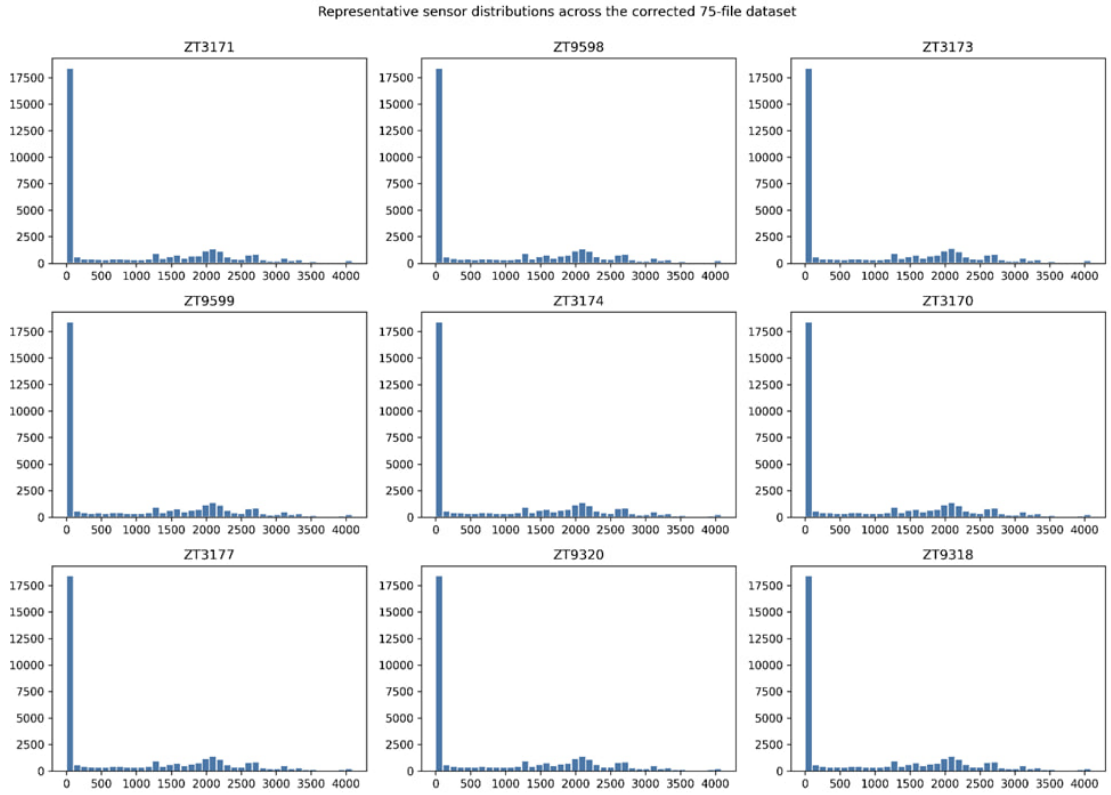


Figure 3.3: sensor distribution across 75-file dataset

Table 3.2: Sensor statistics

Sensor	Mean	Std	Min	Max	Skew	Kurtosis
ZT10080	2122.757	104.3894	2048	2546.5	1.786824	2.167234
ZT10081	2122.720	104.5085	2048	2546.5	1.785840	2.165221
ZT10082	2122.668	104.3264	2048	2545.5	1.785958	2.163634
ZT10083	2122.607	104.3600	2048	2546.0	1.785441	2.162539
ZT10214	2123.305	104.9772	2048	2549.0	1.775662	2.124913
ZT10215	2123.317	105.0326	2048	2550.0	1.775532	2.125499
ZT10216	2123.337	105.2527	2048	2551.0	1.777138	2.129912
ZT10217	2123.597	105.2166	2048	2552.0	1.779473	2.136836
ZT3144	96.22321	138.5556	0	660.0	1.799524	2.203240
ZT3145	96.11459	138.3529	0	660.0	1.799122	2.200559

for simplicity, only the first 10 sensors were listed

3.4.2 Temporal Patterns and Daily Profiles

Sensor signals exhibit consistent daily operational structure. A typical day includes (i) a startup phase with rapid transitions during system initialization, (ii) an operational phase with cyclic patterns aligned with driving behavior and traction load variation, and (iii) a shutdown phase with reduced activity.

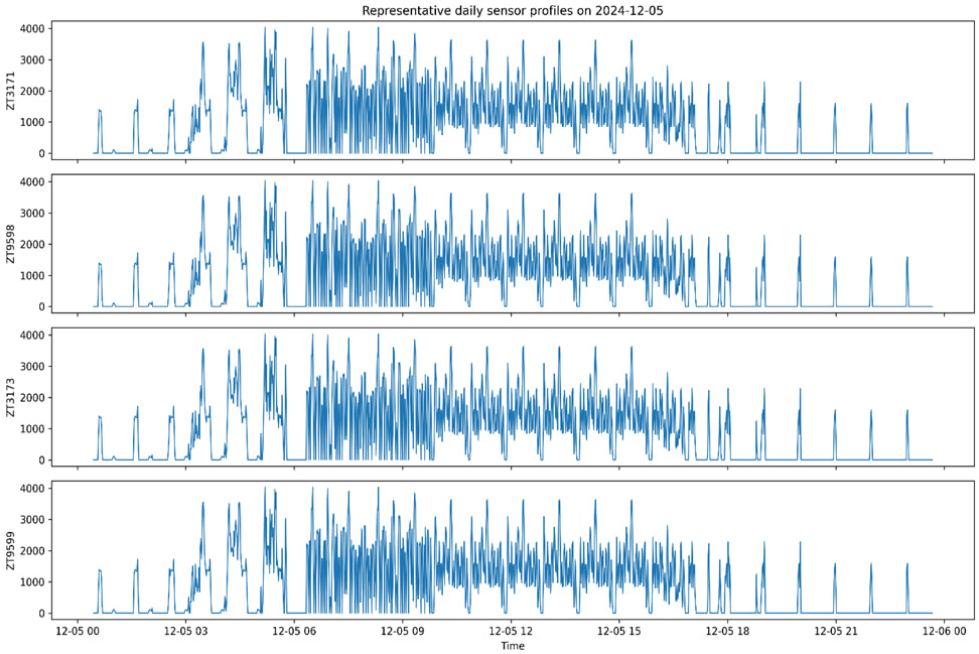


Figure 3.4: daily sensor profiles on 2024-12-05

Cross-day comparison reveals that while the general daily structure is consistent, significant variation exists in signal amplitudes and patterns across different months. This variation may reflect seasonal effects (ambient temperature, passenger load), route differences, or gradual system evolution.

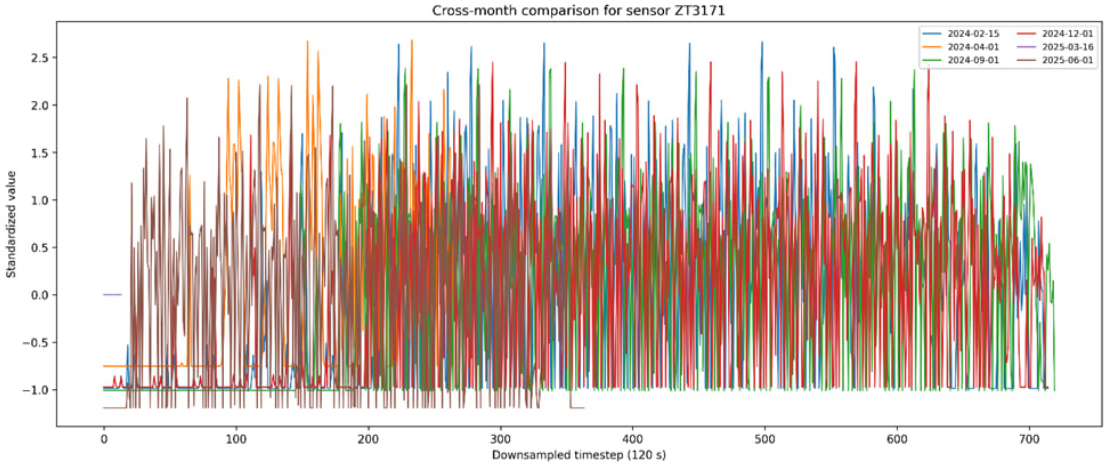


Figure 3.5: cross-month comparison for sensor ZT31171

3.4.3 Sensor Correlations

Many sensors are physically coupled and therefore exhibit correlated behavior. For example, higher motor current typically increases thermal load, producing positive correlation between current-related and temperature-related sensors.

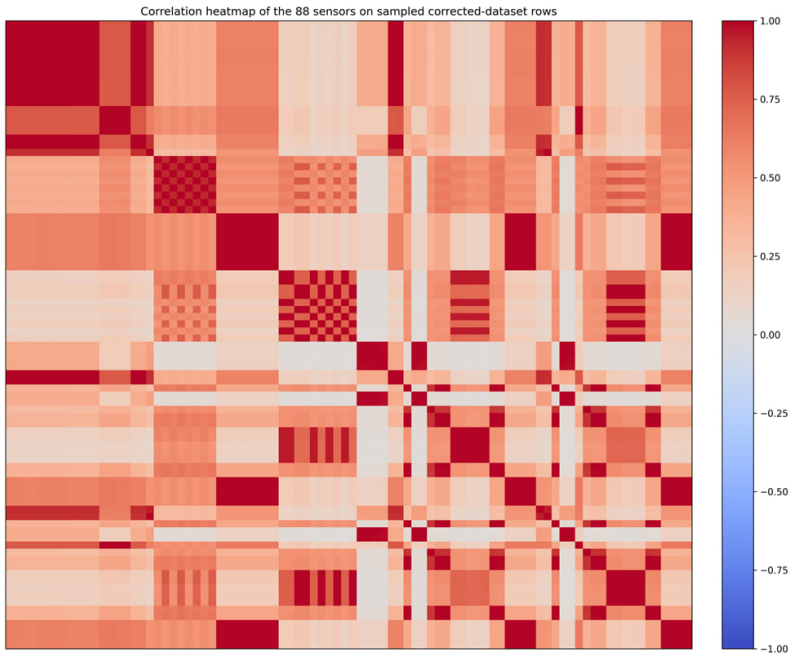


Figure 3.6: correlation heatmap of 88 sensors

The resulting correlation structure reveals clustered sensor groups, indicating that the array captures multiple interrelated physical processes. This multivariate structure is important for anomaly detection because models must learn both individual sensor behavior and cross-sensor dependencies.

3.4.4 Data Completeness and Recording Characteristics

Data completeness was assessed by reindexing each daily dataset to a strict one-second grid and inspecting timestamps and sensor values for missingness. The dataset was found to be exceptionally clean: across all 45 daily datasets, **no missing timestamps** were detected within each recording interval and **no sensor-level NaN values** were observed.

While intra-day recordings are complete, recording duration varies significantly across days. Thus, the main source of heterogeneity is **recording length across days**, rather than missing data within a day.

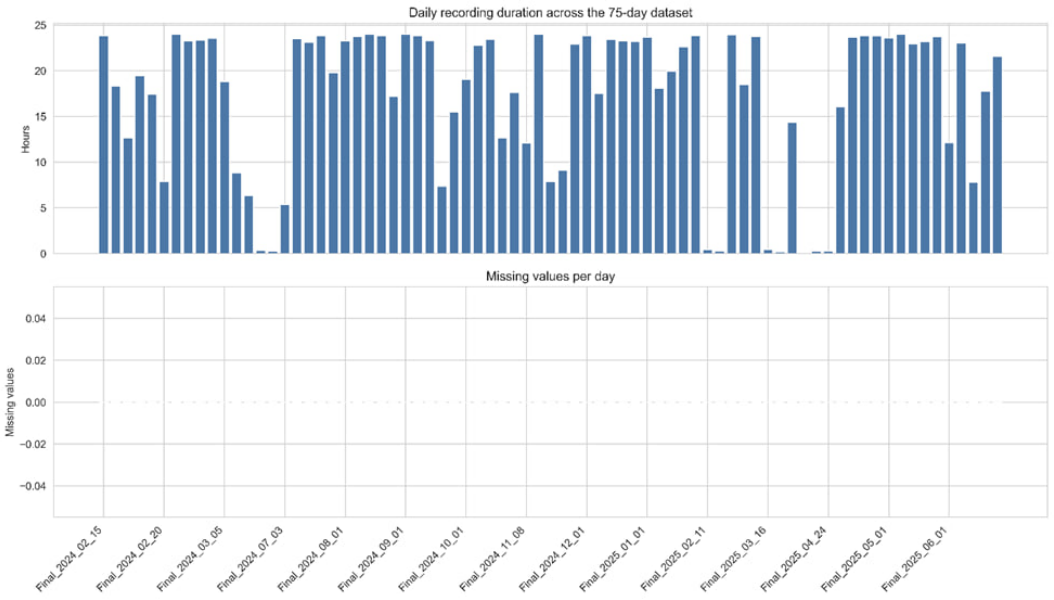


Figure 3.7: Daily recording duration across 75-day dataset

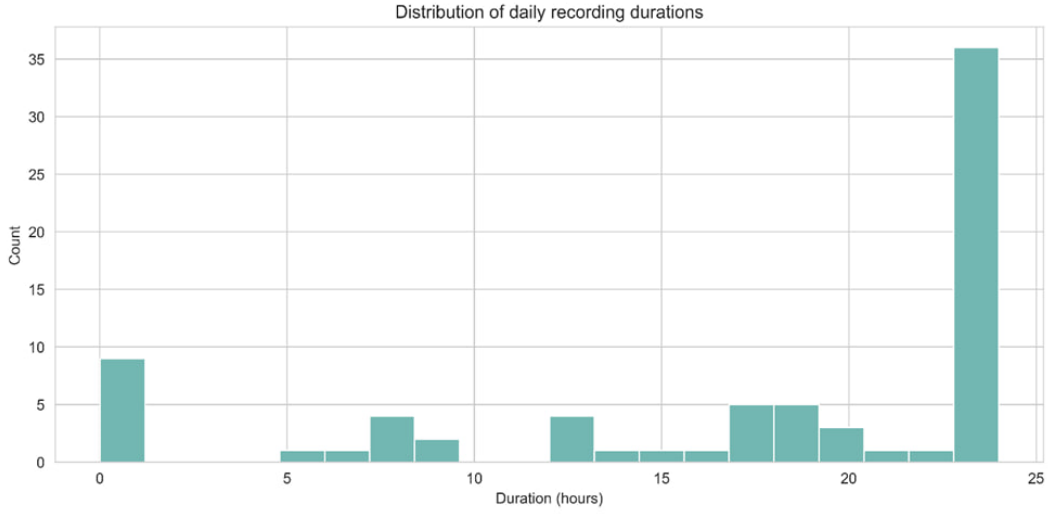


Figure 3.8: Distribution of daily recording durations

3.5 Final Preprocessed Dataset

After temporal verification and normalization, each daily dataset is represented as a clean multivariate time series sampled at a uniform resolution of one data point per second. The pipeline ensures consistent temporal structure and comparable sensor scaling, producing stable inputs for windowing, model training, and anomaly scoring.

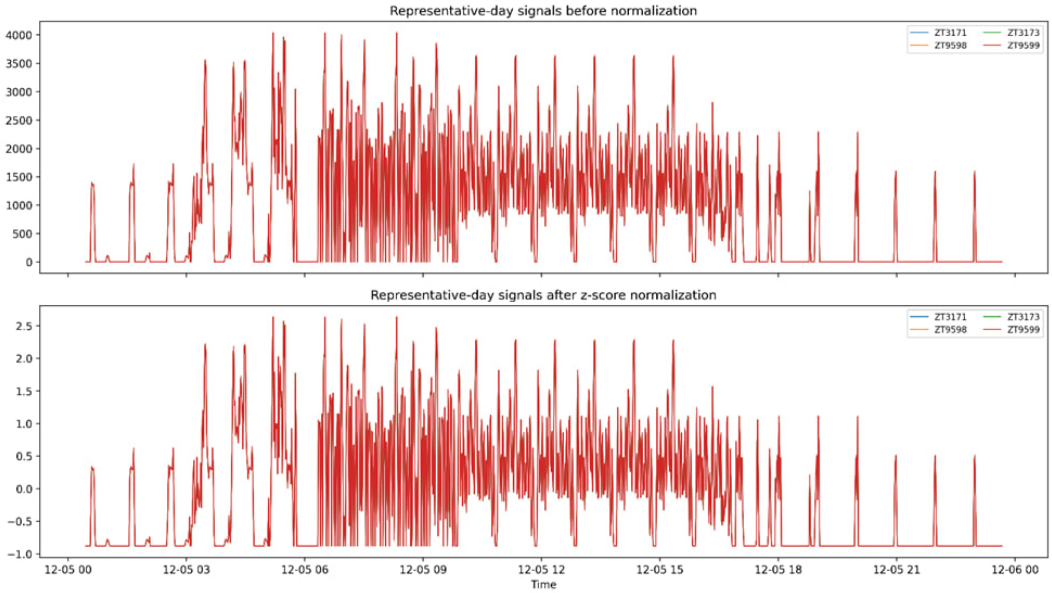


Figure 3.9: Day signals before and after Normalization

By applying an identical preprocessing pipeline across all datasets, the study ensures that performance differences observed in later chapters arise from modeling assumptions rather than preprocessing variability.

Chapter 4

Anomaly Detection Models

This chapter presents the four unsupervised anomaly detection models evaluated in this study. All models learn representations of normal system behavior from multivariate time-series sensor data and assign anomaly scores that quantify deviation from learned normal patterns. The models are grouped into two paradigms: deterministic reconstruction-based autoencoders (Sections 4.2–4.3) and probabilistic generative models (Sections 4.4–4.5).

4.1 Window-Based Input Formulation

To ensure a fair and consistent comparison, all models operate on fixed-length windowed segments of the preprocessed sensor data.

Let $x_t \in \mathbb{R}^D$ denote the sensor reading vector at time t, where $D = 88$ is the number of sensors. For a fixed window length W, each input window consists of W consecutive observations.

$$x_t \in \mathbb{R}^D \tag{4.1}$$

$$D = 88 \tag{4.2}$$

$$X_t = [x_{t-W+1}, x_{t-W+2}, \ldots, x_t] \in \mathbb{R}^{W \times D} \tag{4.3}$$

Each model outputs a scalar anomaly score s_t for the input window, where larger values indicate stronger deviation from learned normal behavior. All models are trained in a fully unsupervised manner using data assumed to be predominantly normal.

Windows are generated using a sliding window with stride 1 s, so consecutive windows overlap by W - 1 timesteps. This dense overlap provides high temporal resolution anomaly scoring and produces smoother score trajectories.

All four models in this chapter share the same underlying idea, which can be stated in one sentence: *learn what normal looks like, and measure how far the present is from it.* Each model is trained exclusively on data assumed to be predominantly normal (Assumption 1, Section 1.4.2). During training, the model builds an internal, compressed description of typical system behavior. At detection time, every new window is confronted with this internal description, and the mismatch between the two becomes the anomaly score. What distinguishes the four models is *how* they build this internal description and *what kind of mismatch* they measure.

The two deterministic models (LSTM-AE and Conv-AE) measure mismatch as a reconstruction error—literally, how badly the model fails to redraw the input signal from its compressed summary. The two probabilistic models (VAE-Prob and RDSSM) instead measure mismatch as statistical implausibility—how unlikely the observed window is under a learned probability distribution of normal behavior. This distinction may seem subtle, but it has concrete consequences: a signal can be perfectly plausible in amplitude yet statistically abnormal in its temporal structure, and only a probabilistic model can flag such a case. The remainder of this chapter presents each model in turn, and Table 4.1 summarizes their differences.

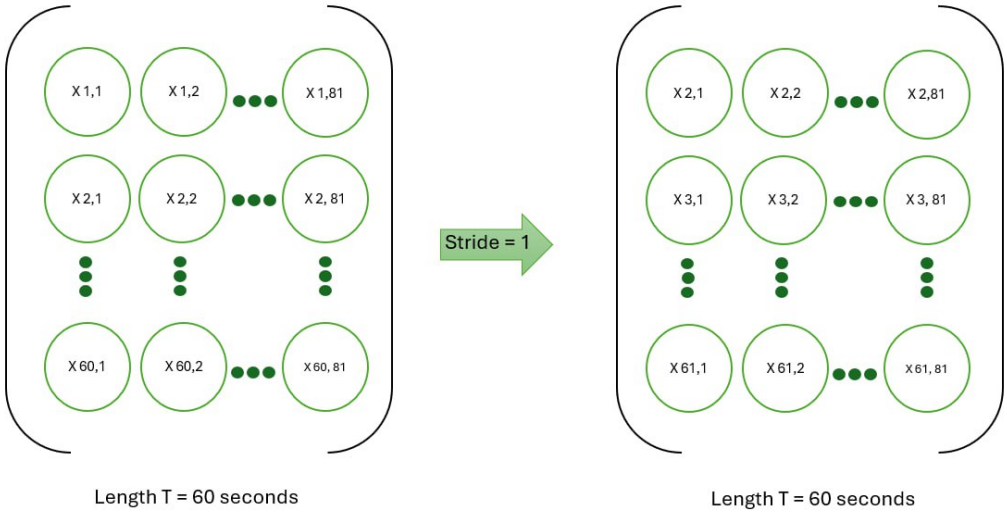


Figure 4.1: Windowing

Windows are generated using a sliding window with stride 1 s, so consecutive windows overlap by W - 1 timesteps. This dense overlap provides high temporal resolution anomaly scoring and produces smoother score trajectories.

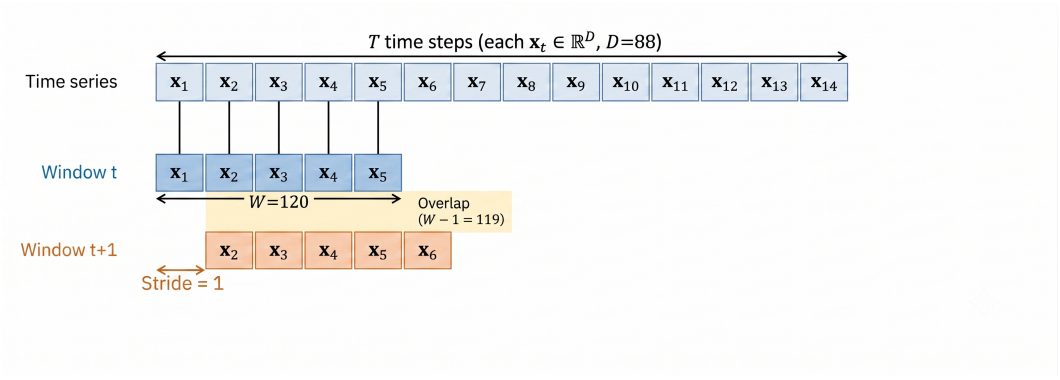


Figure 4.2: T time steps

4.2 LSTM Autoencoder (LSTM-AE)

4.2.1 Architecture

The LSTM Autoencoder uses a recurrent encoder–decoder architecture designed to capture temporal dependencies in sequential sensor signals.

Encoder. The encoder processes the input sequence step-by-step and maintains a hidden state and cell state.

$$h_i, c_i = \mathrm{LSTM}_{\mathrm{enc}}(x_i, h_{i-1}, c_{i-1}), \qquad i = t - W + 1, \ldots, t$$

$$h_i \in \mathbb{R}^H$$

After the final step, the last hidden state is projected into a latent representation:

$$z = W_z h_t + b_z \in \mathbb{R}^L$$

Decoder. The latent vector is projected back to initialize the decoder, and the decoder generates reconstructions across all time steps.

$$\tilde{h}_0 = W_d z + b_d$$

$$\tilde{h}_i, \tilde{c}_i = \mathrm{LSTM}_{\mathrm{dec}}(\tilde{h}_{i-1}, \tilde{c}_{i-1})$$

$$\hat{x}_i = W_o \tilde{h}_i + b_o$$

The reconstructed window is:

$$\hat{X}_t = [\hat{x}_{t-W+1}, \ldots, \hat{x}_t]$$

Hyperparameters. The model uses $H = 64$, $L = 32$, and one LSTM layer for both encoder and decoder.

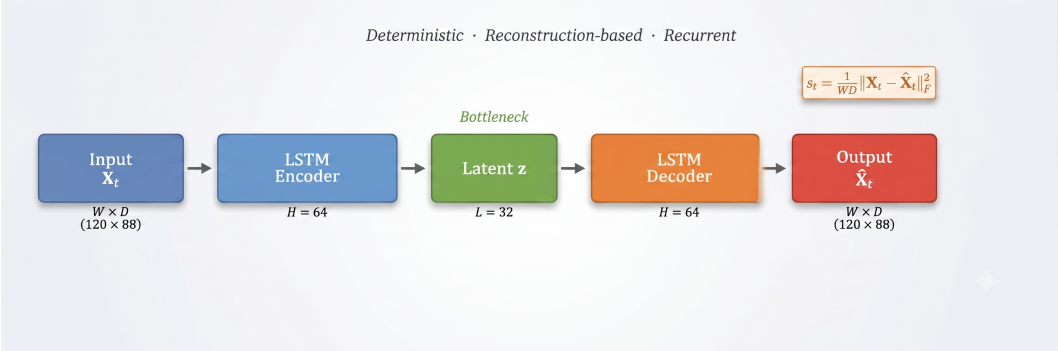


Figure 4.3: LSTM-AE Architecture

Operating principle. An autoencoder is a neural network trained to perform an apparently trivial task: reproduce its own input at its output. The task becomes meaningful because the network is forced to pass the information through a narrow internal "bottleneck", a latent vector much smaller than the input itself [23]. To reconstruct a two-minute window of 88 sensors from a bottleneck of only 32 numbers, the network must discover and exploit the regularities of normal operation: which sensors move together, which patterns follow which, what a typical acceleration or braking phase looks like. Anything that fits these learned regularities can be compressed and reconstructed well; anything that violates them cannot. The reconstruction error therefore acts as a natural anomaly detector: normal windows reconstruct almost perfectly (low score), while abnormal windows come back distorted (high score) .

The LSTM variant of the autoencoder processes the window *sequentially*, one second at a time, using Long Short-Term Memory cells. An LSTM cell maintains an internal memory that is updated at each time step, allowing the network to remember information from many seconds earlier and use it to interpret the present. This gives the LSTM-AE its defining strength: sensitivity to the order of events and to long-range temporal dependencies. A window whose individual values are all plausible, but whose temporal sequence is wrong (for example, a braking pattern appearing where an acceleration pattern is expected), will produce a high reconstruction error even though no single value is extreme.

Strengths and limitations. The LSTM-AE offers stable training, directly interpretable scores (a high score can be traced back to the poorly reconstructed signal segment), and strong sensitivity to gradual temporal distortions [24]. Its main limitations are the higher computational cost of sequential processing and the absence of any explicit notion of uncertainty: the model produces a single deterministic reconstruction and cannot distinguish "this deviation is within normal

random

4.2.2 Training Objective

The model is trained by minimizing mean squared reconstruction error over the full window.

$$L_{\text{LSTM-AE}} = \frac{1}{WD} \sum_{i=1}^{W} \sum_{d=1}^{D} (x_{i,d} - \hat{x}_{i,d})^2$$

4.2.3 Anomaly Score

The anomaly score is defined as reconstruction error.

$$s_t = \frac{1}{WD} \left\| X_t - \hat{X}_t \right\|_F^2$$

4.3 Convolutional Autoencoder (Conv-AE)

4.3.1 Architecture

The Convolutional Autoencoder replaces recurrence with one-dimensional temporal convolution, emphasizing local pattern extraction.

For Conv1d processing, the input is transposed to (B, D, W).

Encoder. Three convolutional blocks progressively extract and compress features.

$$F_1 = \text{ReLU}\Big(\text{BN}\Big(\text{Conv1d}_{128,k=7}(X^T)\Big)\Big)$$

$$F_2 = \text{ReLU}(\text{BN}(\text{Conv1d}_{64,k=5}(F_1)))$$

$$Z = \text{ReLU}(\text{BN}(\text{Conv1d}_{C_L,k=3}(F_2)))$$

$$C_L = 32$$

Decoder. A symmetric decoder reconstructs the input using the concept of transposed convolution. In implementation, this may be realized with ConvTranspose1d or with upsampling followed by Conv1d.

$$\hat{F}_2 = \mathrm{ReLU}(\mathrm{BN}(\mathrm{Conv1d}_{64,k=3}(Z)))$$

$$\hat{F}_1 = \mathrm{ReLU}\Big(\mathrm{BN}\Big(\mathrm{Conv1d}_{128,k=5}(\hat{F}_2)\Big)\Big)$$

$$\hat{X}^T = \mathrm{Conv1d}_{D,k=7}(\hat{F}_1)$$

Finally, transpose back to (B, W, D).

Operating principle. The Convolutional Autoencoder follows the same compress-and-reconstruct principle as the LSTM-AE, but replaces sequential recurrence with one-dimensional convolution [25]. A convolutional filter is a small pattern template, here spanning 7, 5, or 3 seconds, that slides along the time axis and responds strongly wherever the signal locally matches the template. By stacking three convolutional layers, the network learns a hierarchy of local patterns: the first layer captures elementary shapes such as rises, drops, and oscillations, and deeper layers combine them into more complex short-term motifs. The decoder mirrors this process to rebuild the signal.

The practical consequence of this design is a different *inductive bias* – a different built-in preference about what matters in the data. Where the LSTM-AE emphasizes the long-range order of events, the Conv-AE emphasizes *local shape*. It excels at detecting abrupt, localized disturbances: sudden spikes, sharp level shifts, or short bursts of irregularity that break the expected local waveform [26]. Because convolutions process all time steps in parallel rather than one by one, the Conv-AE is also computationally efficient, which matters when scoring hundreds of millions of data points .

Strengths and limitations. The Conv-AE is fast, robust to train, and highly responsive to abrupt local changes. Its limitation is the mirror image of its strength: with filters spanning only a few seconds, it has a limited view of long-range temporal structure, and slowly developing anomalies—such as a drift unfolding over many

minutes—may remain invisible to it, since every local segment of a slow drift still looks locally normal.

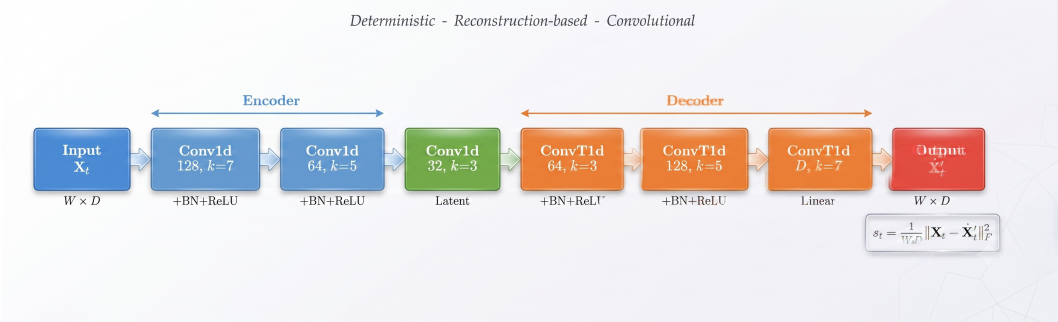


Figure 4.4: Convolutional-AE Architecture

4.3.2 Training Objective and Anomaly Score

The Conv-AE is trained with MSE and scored via reconstruction error, identical in form to the LSTM-AE.

$$L_{\text{Conv-AE}} = s_t = \frac{1}{WD} \left\| X_t - \hat{X}_t \right\|_F^2$$

4.4 Variational Autoencoder (VAE-Prob)

4.4.1 Motivation

Deterministic autoencoders output point estimates and do not represent uncertainty explicitly. This limits their ability to distinguish stochastic variation from meaningful abnormality. The VAE introduces a probabilistic latent variable model to enable uncertainty-aware scoring [27].

Operating principle. The Variational Autoencoder keeps the encoder–decoder structure but changes the nature of the bottleneck in a fundamental way: instead of mapping each input window to a single fixed point in latent space, the encoder maps it to a probability distribution—concretely, a Gaussian described by a mean vector μ and a variance vector σ^2 [13]. A latent sample z is then drawn from this distribution and passed to the decoder. During training, a regularization term (the Kullback–Leibler divergence) gently pushes all these per-window distributions

toward a common standard normal prior, which organizes the latent space into a smooth, continuous map of normal behavior in which similar operating conditions occupy neighboring regions

This probabilistic construction changes what the anomaly score means. The score is no longer just "how badly was the input redrawn," but an approximation of "how unlikely is this window under the learned model of normality" (the negative evidence lower bound, combining reconstruction quality and latent plausibility) [27]. An observation can therefore be flagged as anomalous not only when its values are distorted, but also when its overall statistical configuration—the joint pattern across the 88 sensors—falls outside the region of latent space that normal data occupy. In addition, the stochastic sampling acts as a built-in tolerance for noise: small random fluctuations of the input move the latent sample only slightly within its distribution and barely affect the score, which makes the VAE naturally more robust to harmless measurement noise than its deterministic counterparts.

Strengths and limitations. The VAE-Prob brings uncertainty awareness and a principled probabilistic interpretation of the score, improving robustness in noisy conditions and enabling the detection of distributional anomalies that pure reconstruction error can miss. Its costs are a more delicate training procedure, balancing reconstruction against regularization through the weight β, and, in the architecture adopted here, the absence of explicit temporal modeling: the window is flattened before encoding, so the model sees the joint statistical pattern of the window but has no dedicated mechanism for the *order* of events inside it.

4.4.2 Architecture

The VAE uses an MLP encoder–decoder on flattened windows.

Encoder. The window is flattened:

$$x_{\text{flat}} \in \mathbb{R}^{W \cdot D}$$

The encoder produces parameters of a diagonal Gaussian posterior:

$$h = \text{ReLU}(W_2\text{ReLU}(W_1 x_{\text{flat}} + b_1) + b_2)$$

$$\mu = W_\mu h + b_\mu \in \mathbb{R}^L$$

$$\log(\sigma^2) = W_\sigma h + b_\sigma \in \mathbb{R}^L$$

$$L = 16$$

Reparameterization trick.

$$z = \mu + \sigma \odot \epsilon$$

$$\epsilon \sim \mathcal{N}(0, I)$$

Decoder.

$$\hat{x}_{\text{flat}} = \text{MLP}_{\text{dec}}(z) \in \mathbb{R}^{W \cdot D}$$

$$\hat{X}_t \in \mathbb{R}^{W \times D}$$

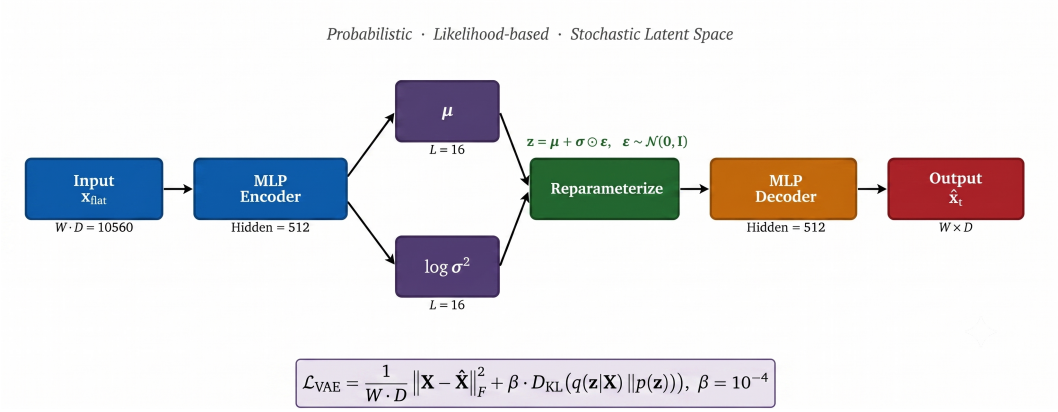


Figure 4.5: Variational-AE Architecture

4.4.3 Training Objective

The VAE is trained by optimizing an ELBO-based loss consisting of reconstruction plus KL regularization.

$$L_{\text{VAE}} = \frac{1}{WD} \left\| X_t - \hat{X}_t \right\|_F^2 + \beta D_{\text{KL}}(q(z|X_t) \,\|\, p(z))$$

with:

$$q(z|X_t) = \mathcal{N}\Big(\mu, \text{diag}(\sigma^2)\Big)$$

$$p(z) = \mathcal{N}(0, I)$$

Analytical KL for diagonal Gaussians (posterior vs standard normal):

$$D_{\text{KL}} = -\frac{1}{2} \sum_{l=1}^{L} \Big(1 + \log(\sigma_l^2) - \mu_l^2 - \sigma_l^2\Big)$$

The weighting factor is:

$$\beta = 10^{-4}$$

4.4.4 Anomaly Score

The anomaly score is taken as the negative ELBO proxy (reconstruction + weighted KL):

$$s_t = \frac{1}{WD} \left\| X_t - \hat{X}_t \right\|_F^2 + \beta D_{\text{KL}}$$

4.5 Recurrent Deep State-Space Model (RDSSM)

4.5.1 Motivation

Unlike autoencoders that learn a static mapping, RDSSM explicitly models the evolution of latent system state through stochastic dynamics. This enables detection of anomalies that arise from deviations in temporal dynamics rather than only signal amplitude.

Operating principle. The Recurrent Deep State-Space Model is the most structured of the four architectures, and its design borrows a classical idea from control engineering: at every instant, the system possesses a hidden internal *state*—an unobserved summary of its true condition—and the sensor readings are noisy observations *emitted* by that state [15]. The model learns two things jointly. The first is a **transition model** (the "prior"): given the current latent state, what should the next state be? This is the model's internal expectation of how the system evolves from one second to the next. The second is an **inference model** (the "posterior"): given the actual sensor readings just observed, what is the latent state most likely to be right now? Both components are parameterized by recurrent GRU networks, and both describe the state not as a point but as a probability distribution, so uncertainty is carried through time [28].

Anomaly detection then emerges from the *disagreement between expectation and observation*. At each time step, the model predicts where the system should go (prior) and then corrects this prediction using what the sensors actually report (posterior). When the system behaves normally, prediction and correction agree, and the observations are assigned high likelihood. When the underlying dynamics change—a regime shift, an unusual transition, the onset of a fault—the observations become improbable under the predicted state evolution, the prior–posterior disagreement grows, and the score rises sharply. This is why the RDSSM is particularly effective at detecting anomalies of *dynamics*: cases where the issue is not any individual sensor value, but the way the system moves from one condition to the next.

Strengths and limitations. The RDSSM combines explicit temporal modeling with explicit uncertainty modeling, making it the most complete of the four architectures and, as Chapter 6 shows, the strongest performer on the controlled benchmarks. Its costs are the highest architectural complexity of the four models, a training objective with more interacting components, and an anomaly score living on a very small numerical scale due to its likelihood-based formulation, which makes careful, model-specific threshold calibration essential.

4.5.2 Architecture

RDSSM includes a **prior/transition** model and a **posterior/inference** model.

Prior (transition).

$$h_i^{(\text{prior})} = \text{GRU}_{\text{prior}}\left(z_{i-1}, h_{i-1}^{(\text{prior})}\right)$$

$$\mu_i^{(\text{prior})} = W_\mu^p h_i^{(\text{prior})}$$

$$\log\left(\left(\sigma_i^{(\text{prior})}\right)^2\right) = W_\sigma^p h_i^{(\text{prior})}$$

Posterior (inference).

$$\tilde{h}_i = \text{ReLU}(W_{\text{in}} x_i)$$

$$h_i^{(\text{post})} = \text{GRU}_{\text{post}}\left(\tilde{h}_i, h_{i-1}^{(\text{post})}\right)$$

$$\mu_i^{(\text{post})} = W_\mu^q h_i^{(\text{post})}$$

$$\log\left(\left(\sigma_i^{(\text{post})}\right)^2\right) = W_\sigma^q h_i^{(\text{post})}$$

Sampling.

$$z_i = \mu_i^{(\text{post})} + \sigma_i^{(\text{post})} \odot \epsilon_i$$

$$\epsilon_i \sim \mathcal{N}(0, I)$$

Emission / decoding.

$$\hat{x}_i = \text{MLP}_{\text{dec}}(z_i) = W_3 \text{ReLU}(W_2 z_i + b_2) + b_3$$

Hyperparameters.

- $L_z = 16$, $H = 64$, decoder hidden dimension $= 128$.

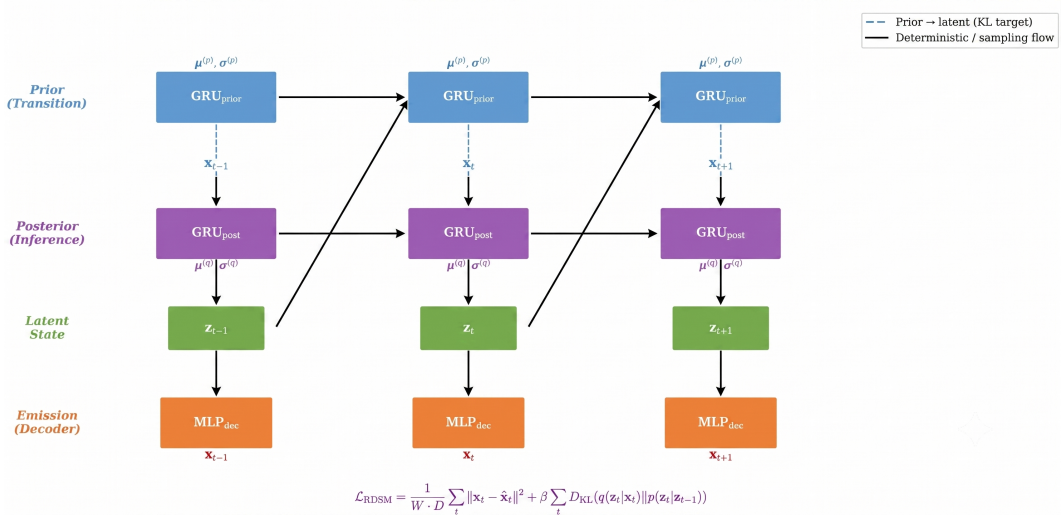


Figure 4.6: RDSSM Architecture

4.5.3 Training Objective

RDSSM is trained using a variational objective combining reconstruction and a temporal KL term between posterior and prior at each time step.

$$L_{\text{RDSSM}} = \frac{1}{WD}\sum_{i=1}^{W}\|x_i - \hat{x}_i\|^2 + \beta\frac{1}{W}\sum_{i=1}^{W} D_{\text{KL}}(q(z_i|x_i)\,\|\,p(z_i|z_{i-1}))$$

KL between two diagonal Gaussians:

$$D_{\text{KL}}\Big(\mathcal{N}(\mu_q,\sigma_q^2)\,\|\,\mathcal{N}(\mu_p,\sigma_p^2)\Big) = \frac{1}{2}\sum_{l=1}^{L_z}\left(\frac{\sigma_{q,l}^2}{\sigma_{p,l}^2} + \frac{(\mu_{p,l}-\mu_{q,l})^2}{\sigma_{p,l}^2} + \log\frac{\sigma_{p,l}^2}{\sigma_{q,l}^2} - 1\right)$$

Weight:

$$\beta = 10^{-4}$$

4.5.4 Anomaly Score

The anomaly score is derived from negative log-likelihood; in practice it is approximated by the same terms used in training (reconstruction + KL).

$$s_t = -\frac{1}{W}\sum_{i=1}^{W}\log p(x_i|z_i)$$

Table 4.1: Model architecture comparison summary

Property	**LSTM-AE**	**Conv-AE**	**VAE-Prob**	**RDSSM**
Paradigm	Deterministic	Deterministic	Probabilistic	Probabilistic
Temporal modeling	Recurrent (long-range)	Convolutional (local)	None (MLP)	Recurrent (state-space)
Latent space	Deterministic vector	Deterministic features	Stochastic distribution	Stochastic, time-varying
Uncertainty modeling	No	No	Yes	Yes
Scoring mechanism	Reconstruction error	Reconstruction error	Negative ELBO	Negative log-likelihood
Hidden dim	64	128/64/32	512	64
Latent dim	32	32 channels	16	16
Strengths	Memory, stability	Local patterns, efficient	Uncertainty-aware	Dynamic latent modeling
Limitations	No uncertainty	Limited long-range	No temporal structure	Higher complexity

Chapter 5

Experimental Setup

This chapter describes the experimental pipeline used to train, evaluate, and compare the four anomaly detection models. All experiments are conducted within a unified and fully reproducible framework applied consistently across **74 daily datasets** (one day was excluded due to insufficient recording duration). This design ensures that observed differences primarily reflect model architecture and probabilistic assumptions rather than variations in data handling or evaluation procedures.

Read together, the four models form a two-by-two design that is the backbone of the entire experimental comparison. Along one axis, they differ in **how they model time**: the LSTM-AE and the RDSSM process the sequence step by step and capture long-range order, the Conv-AE captures local shape, and the VAE-Prob captures the joint statistical pattern of the window without explicit ordering. Along the other axis, they differ in **whether they model uncertainty**: the two autoencoders produce deterministic point reconstructions, while the VAE-Prob and the RDSSM describe normal behavior as a probability distribution and score observations by their plausibility. Every combination of these two design choices is represented, which is precisely what allows Chapter 6 to attribute observed behavioral differences to specific modeling assumptions rather than to incidental implementation details.

5.1 Experimental Pipeline Overview

The experimental workflow follows a structured sequence:

1. Preprocessed continuous time-series data from Chapter 3 are segmented into overlapping windows.
2. Data are split into training and validation sets.
3. Models are trained in an unsupervised manner on the training set.
4. Anomaly scores are computed for all windows (training + validation).
5. Percentile-based thresholding is applied using training-set statistics.
6. Anomalous time steps are aggregated into events.
7. Unsupervised quality metrics are computed.
8. Cross-model comparative analysis is performed.

This standardized pipeline ensures methodological fairness and supports transparency and reproducibility.

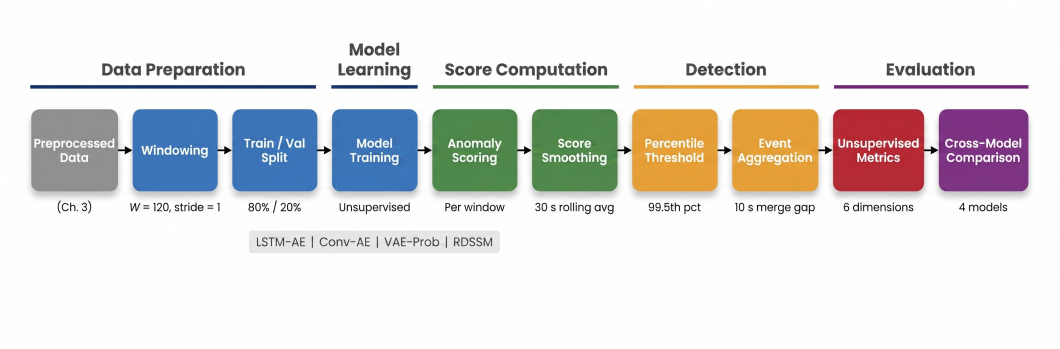


Figure 5.1: Experimental pipeline overview

5.2 Time-Series Windowing Strategy

Deep learning architectures require fixed-length inputs. Each continuous daily time series is therefore segmented into overlapping windows prior to model training.

Let W denote window length and Δ the stride. At each time step t, the windowed input is defined as:

$$X_t = [x_{t-W+1}, \ldots, x_t] \in \mathbb{R}^{W \times D}$$

This transforms a long sequence into a dataset of temporally ordered subsequences.

Window length. $W = 120$ s (2 minutes) is selected to balance temporal context and computational efficiency. This duration captures meaningful short-term dynamics (e.g., rapid regime changes and local cycles) without excessive computational cost.

Stride. $\Delta = 1$ s, so consecutive windows overlap by:

$$W - 1 = 119$$

This dense overlap provides high temporal resolution scoring, smooth score trajectories, and precise localization of deviations.

Window count example. For a recording of length T seconds, the number of windows is approximately $T - W + 1$ when $\Delta = 1$. For a typical day of about 65,000 seconds:

$$N_{\text{windows}} = T - W + 1$$

$$N_{\text{windows}} = 65000 - 120 + 1 = 64881$$

5.3 Data Splitting Strategy

To evaluate generalization while preserving temporal structure, a **temporal** train–validation split is performed within each daily dataset.

Split method. The first 80% of time steps are used for training and the remaining 20% are reserved for validation.

$$D_{\text{train}} = \{X_t \mid t \leq 0.8T\}$$

$$D_{\text{val}} = \{X_t \mid t > 0.8T\}$$

where $0.8T$ denotes the timestamp at the 80th percentile of the day's recording.

Rationale. This split mirrors deployment conditions in which a model is trained on historical data and applied to future unseen periods. The training set is assumed to be predominantly normal, consistent with the unsupervised learning assumption.

Small consistency note (recommended wording): In practice, scores are computed per window and later mapped to time steps (via window-to-time alignment or aggregation). Your pipeline description is fully fine; this note just clarifies the unit (window vs time step) if you want extra precision.

Table 5.1: Representative split summary

Split	Approximate Duration	Percentage	Approximate Windows
Training	∼66,897 s (∼18.6 h)	80%	∼66,778
Validation	∼16,725 s (∼4.6 h)	20%	∼16,725
Total	∼83,000 s (∼23.2 h)	100%	∼83,503

5.4 Training Protocol

All models are trained in a strictly unsupervised setting. No anomaly labels are used during optimization.

Optimization. Adam optimizer is used with learning rates:

$$\alpha = 10^{-3} \qquad \text{(LSTM-AE)}$$

$$\alpha = 10^{-3} \qquad \text{(Conv-AE)}$$

$$\alpha = 3 \times 10^{-4} \qquad \text{(VAE-Prob)}$$

$$\alpha = 3 \times 10^{-4} \qquad \text{(RDSSM)}$$

Training duration. All models are trained for 30 epochs. Early stopping based on anomaly labels is not applicable due to label scarcity.

Batch size. 128 windows per batch.

Training scope. A global training approach is used: a single model is trained across all training days to learn a general representation of normal behavior under diverse operational conditions.

Device. Training is conducted on CUDA-enabled GPUs to handle the large dataset efficiently.

Table 5.2: Training hyperparameters

Parameter	LSTM-AE	Conv-AE	VAE-Prob	RDSSM
Optimizer	Adam	Adam	Adam	Adam
Learning rate	10^{-3}	10^{-3}	3×10^{-4}	3×10^{-4}
Batch size	128	128	128	128
Window size (W)	120 s	120 s	120 s	120 s
Stride (Δ)	1 s	1 s	1 s	1 s
β (KL weight)	–	–	10^{-4}	10^{-4}

5.5 Anomaly Score Computation

Each model outputs a scalar anomaly score per window, measuring deviation from learned normal behavior.

Deterministic models (LSTM-AE, Conv-AE). Reconstruction-based anomaly score:

$$S_t = \frac{1}{WD} \left\| X_t - \hat{X}_t \right\|_F^2$$

Probabilistic models (VAE-Prob, RDSSM). The anomaly score is derived from negative ELBO (or negative log-likelihood proxy), combining reconstruction and a KL regularization term:

$$S_t = -\text{ELBO}(X_t) = \text{Reconstruction Loss} + \beta D_{\text{KL}}$$

Score smoothing. A 30-second rolling average is applied to reduce high-frequency noise:

$$\bar{S}_t = \frac{1}{30} \sum_{i=t-29}^{t} S_i$$

5.6 Thresholding Strategy

Because reliable ground-truth labels are unavailable, a percentile-based thresholding approach converts continuous scores into binary anomaly decisions.

For each model and each day, the threshold τ is defined as the **99.5th percentile** of training scores:

$$\tau = Q_{0.995}(\{S_t \mid t \in D_{\text{train}}\})$$

A time index is flagged as anomalous when its score exceeds the threshold:

$$a_t = \begin{cases} 1, & S_t > \tau \\ 0, & \text{otherwise} \end{cases}$$

This strategy avoids manual tuning, adapts to each model's score scale, and ensures consistent methodology across models and days.

Interpretation note. By design, approximately 0.5% of training points/windows exceed the threshold. Therefore, the raw number of flagged anomalies is largely controlled by threshold choice and is not, by itself, a quality indicator.

5.7 Event-Level Aggregation

Point-wise anomaly flags can produce isolated spikes that do not correspond to meaningful operational deviations. To obtain a more interpretable representation, consecutive anomalous time indices are grouped into contiguous **anomaly events**, with short gaps tolerated to merge nearby detections.

An anomaly event E_k is defined as a maximal contiguous set of anomalous time indices, allowing gaps up to 10 seconds:

$$E_k = \{t_{\text{start}}^{(k)}, \ldots, t_{\text{end}}^{(k)}\} \qquad \text{where } a_t = 1 \text{ for all } t \in E_k$$

with gaps ≤ 10 *s merged into the same event*

For each event, the following attributes are recorded:

$$\text{Duration}_k = t_{\text{end}}^{(k)} - t_{\text{start}}^{(k)} + 1$$

5.8 Unsupervised Evaluation Metrics

In the absence of ground truth labels, evaluation uses unsupervised metrics that quantify the internal structure and stability of anomaly score outputs and detected events.

5.8.1 Score Separability

Max-over-median ratio:

$$R_{\text{sep}} = \frac{\max_t S_t}{\text{median}_t S_t}$$

P99.9-over-P95 tail ratio:

$$R_{\text{tail}} = \frac{Q_{0.999}}{Q_{0.95}}$$

Higher values typically indicate stronger separation and heavier tails in the anomaly score distribution.

5.8.2 Event-Level Metrics

- **Event count:** number of detected events per day
- **Event duration:** median and $Q_{0.95}$ of event durations
- **Mean Time Between Events (MTBE):** mean time gap between consecutive events
- **Burstiness:** coefficient of variation of inter-event intervals

5.8.3 Temporal Stability

Fragmentation. Defined as the ratio of the number of events to the number of anomalous time indices:

$$\text{Fragmentation} = \frac{\#\text{events}}{\#\text{anomalous time steps}}$$

Lower values indicate more contiguous detections (less fragmentation).

5.8.4 Threshold Robustness

Robustness is assessed by varying the threshold by multipliers $\{0.8, 0.9, 1.0, 1.1, 1.2\}$ and measuring how smoothly the number of detected anomalies changes. Robust methods typically show gradual changes rather than abrupt jumps.

5.8.5 Percentage of Anomalous Time

$$P_{\text{anom}} = \frac{N_{\text{anom}}}{N_{\text{total}}} \times 100$$

5.9 Cross-Model Comparison Framework

Models are compared along multiple dimensions:

1. Temporal modeling capability
2. Sensitivity to deviations (subtle vs abrupt)
3. Stability of anomaly scores
4. Score distribution properties (scale, tails)
5. Event coherence (contiguous vs fragmented)
6. Threshold robustness

Since score scales differ substantially across deterministic and probabilistic models, comparisons are made relative to each model's own score distribution and threshold. This isolates behavioral differences rather than raw magnitude differences.

5.10 Sensitivity Analysis Design

A sensitivity analysis is used to assess how key hyperparameters influence detection behavior.

1. **Threshold percentile:** $p \in \{99.0, 99.5, 99.9\}$

   evaluated by recomputing thresholds from existing scores; no retraining required

2. **Window size:** $W \in \{60, 120, 240\}$ s

   requires retraining; only baseline $W = 120$ s fully evaluated; expected effects discussed

3. **Latent dimension:** $L \in \{8, 16, 32\}$ for VAE-Prob and RDSSM

   requires retraining; only baseline $L = 16$ fully evaluated; expected effects discussed

The sensitivity variables are summarized as:

- $p \in \{99.0, 99.5, 99.9\}$
- $W \in \{60, 120, 240\}$
- $L \in \{8, 16, 32\}$

Table 5.3: Sensitivity analysis configurations

Parameter	Baseline Value	Alternative Values	Status
Threshold percentile	99.5	99.0, 99.9	Fully evaluated
Window size	120 s	60, 240 s	Baseline only (discussed)
Latent dimension	16	8, 32	Baseline only (discussed)

5.11 Synthetic Anomaly Benchmark Design

Because dense and reliable real anomaly labels are not available in the dataset, the final evaluation framework was extended with a controlled synthetic-anomaly benchmark [29]. This benchmark does not change the unsupervised training protocol: all models are trained on the clean data only, and synthetic events are injected exclusively into the validation portion of each day.[5]

The benchmark introduces five anomaly families designed to reflect qualitatively different sensor failures or abnormal behaviors while remaining compatible with the 120-second multivariate windowing strategy:

1. **Spike pulse:** short transient pulse spanning multiple sensors so that the anomaly perturbs the full window score.
2. **Step shift:** sustained multivariate level offset affecting several correlated channels.
3. **Gradual drift:** progressive trend deviation over a long validation interval.
4. **Variance burst:** localized high-variance noisy segment across multiple sensors.
5. **Stuck sensor:** flatline behavior in which selected channels become frozen at an abnormal offset for an extended duration.

For each evaluable day and for each model, one controlled event of each family is injected into the validation segment using deterministic random seeds to ensure reproducibility. The synthetic days are then split into calibration and test partitions. The calibration partition is used only to select a single threshold multiplier and a minimum sustained anomaly duration for each model, starting from the clean training-only threshold estimated in the main experiments. Final benchmark numbers are reported on the held-out synthetic test partition.

The synthetic benchmark is evaluated with point-level and event-level metrics [5]:

- Precision
- Recall
- F1-score
- AUROC

- AUPRC
- Event detection rate
- Detection delay

This design yields an objective, reproducible complement to the real-data unsupervised metrics while preserving the unsupervised nature of model training and avoiding overly diluted synthetic events

5.12 Improvement Design

The improvement-lab protocol followed these rules:

1. The corrected baseline rerun was frozen as the reference result set.
2. The four baseline models were loaded from their preserved weights.
3. Each model was fine-tuned using a shared denoising strategy on normal training windows only.
4. The same synthetic benchmark design and the same held-out test split were reused.
5. Threshold multiplier and minimum sustained anomaly duration were calibrated only on synthetic calibration days.
6. The final improved results were reported on the same held-out synthetic test cases used for the baseline comparison.

Chapter 6

Experimental Results

This chapter presents the final experimental results obtained after correcting the dataset scope, rerunning the full baseline pipeline, introducing the synthetic-ground-truth benchmark, and performing the improvement-lab study. The chapter preserves the original thesis result structure while adding the new benchmark and the baseline-vs-improved comparison.

6.1 Overview of Evaluation Scope

The final experimental evaluation was conducted across 74 of the 75 daily datasets described in Chapter 3; one day (`Final_2025_03_23`) was excluded due to insufficient recording duration for meaningful 120-second windowed evaluation. The evaluated days span February 2024 to June 2025, ensuring that model behavior was assessed over a diverse range of operational conditions.

Two result layers are now reported:

1. Baseline rerun results: on the corrected 75-day dataset.

2. Improvement-lab results: obtained from the preserved baseline weights after denoising fine-tuning.

6.2 Anomaly Score Behavior

The anomaly scores produced by the four models continue to operate on fundamentally different numerical scales due to their distinct objective functions.

Deterministic models (LSTM-AE and Conv-AE) produce reconstruction-error-based scores that are moderately scaled and vary smoothly over time. These scores directly reflect the magnitude of discrepancy between observed and reconstructed signals.

Probabilistic models (VAE-Prob and RDSSM) produce likelihood-based scores that may exhibit sharper localized variations, particularly when observations deviate from the learned latent dynamics. RDSSM operates on a substantially smaller numerical scale due to its state-space likelihood formulation.

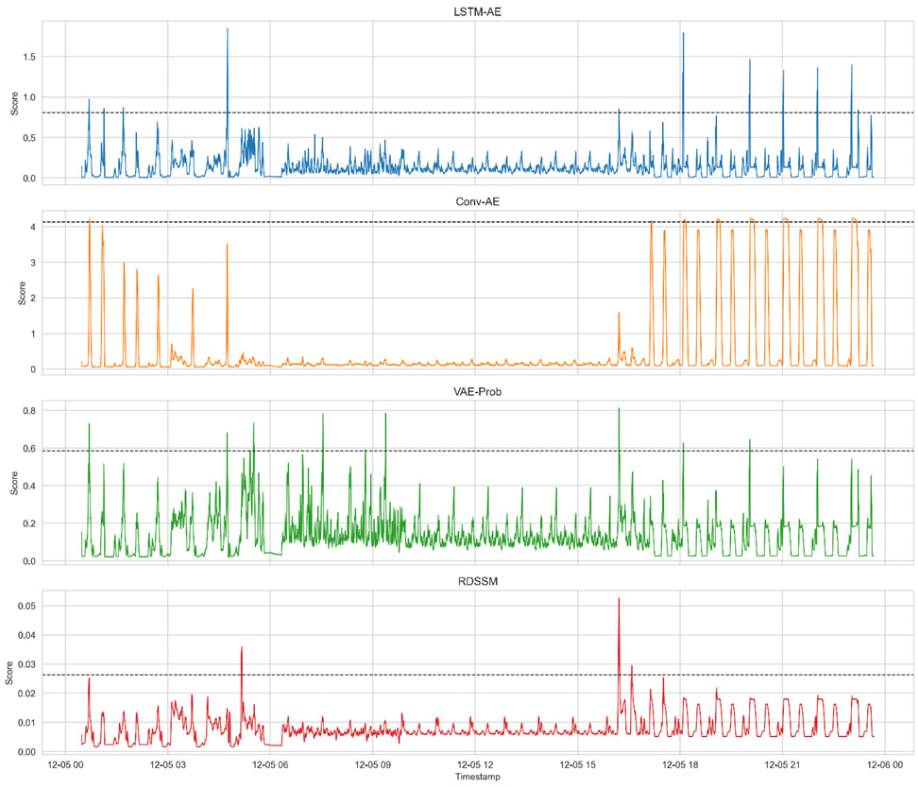


Figure 6.1: Anomaly score behavior on the representative day

A visual comparison of the four score trajectories on the representative day (Figure 6.1) makes the paradigm difference tangible. The LSTM-AE trace evolves

smoothly, rising and falling gradually as the reconstruction quality degrades and recovers; its peaks are broad and easy to associate with extended stretches of unusual signal. The Conv-AE trace is spikier: it stays close to zero for long stretches and then jumps sharply whenever a localized disturbance breaks the expected local waveform, which is exactly the behavior its short convolutional filters predispose it to. The VAE-Prob trace sits between these two extremes — moderately smooth, thanks to the averaging effect of its stochastic latent space, but with more pronounced local structure than the LSTM-AE. The RDSSM trace is the most distinctive: it operates on a numerical scale roughly twenty times smaller than the others, remains very flat during stable operation, and produces narrow, sharp excursions precisely at moments of transition, when the observed dynamics disagree with the model's predicted state evolution. In short, the same day of data, seen through four different models, yields four visibly different "signatures" of abnormality — a first qualitative confirmation that architecture, not data handling, drives detection behavior.

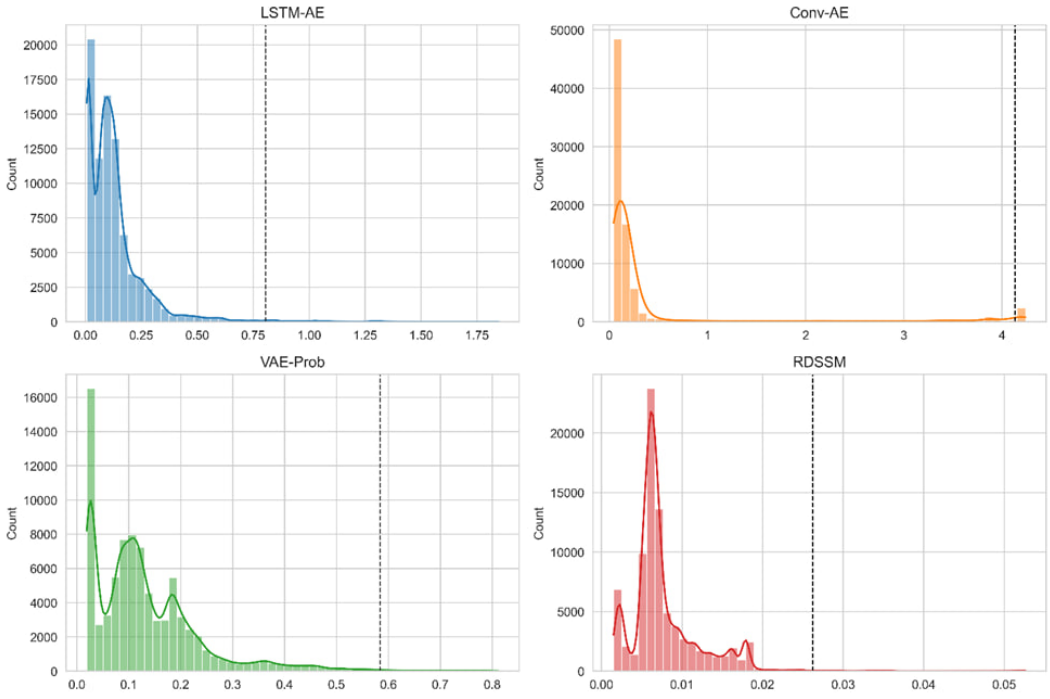


Figure 6.2: Anomaly score distributions with their corresponding anomaly thresholds.

6.3 Threshold Analysis

Using the 99.5th percentile strategy described in Chapter 5, the following median thresholds were observed across the 74 days:

Model	Typical Threshold (Median)
LSTM-AE	0.5925
Conv-AE	0.9257
VAE-Prob	0.6063
RDSSM	0.0209

Table 6.1: Threshold Analysis

These scale differences remain model-specific and do not indicate superiority by themselves. All meaningful comparisons in the thesis are therefore made relative to each model's own score distribution and calibrated operating point.

The spread of the median thresholds — 0.5925 (LSTM-AE), 0.9257 (Conv-AE), 0.6063 (VAE-Prob), and 0.0209 (RDSSM) — deserves a brief comment because it is a frequent source of misinterpretation. The RDSSM threshold is roughly thirty to forty times smaller than the others, yet this says nothing about sensitivity or quality: it is a direct consequence of the likelihood-based formulation of its score, in which even small numerical differences correspond to large differences in probabilistic plausibility. The practical lesson is methodological: anomaly scores from different model families are not interchangeable currencies, and any comparison performed on raw score magnitudes would be meaningless. This is why every comparison in this thesis is made *relative to each model's own score distribution* — through percentiles, ratios, and calibrated operating points — rather than through absolute values.

6.4 Maximum Anomaly Scores

For each model, the day achieving the global maximum anomaly score was identified:

Model	Day of Global Maximum	Threshold	Maximum Score	Max/Threshold Ratio
LSTM-AE	2024-09-02	1.7307	5.6248	3.2501
Conv-AE	2024-09-05	10.1808	10.3672	1.0183
VAE-Prob	2024-09-02	0.748	2.5396	3.395
RDSSM	2024-09-03	0.2313	0.9701	**4.1934**

Table 6.2: Maximum Anomaly score

All four models again concentrate their strongest baseline detections around early September 2024, reinforcing the conclusion that this period contains structurally unusual operational behavior. LSTM-AE and VAE-Prob peak on 2024-09-02, Conv-AE peaks on 2024-09-05, and RDSSM peaks on 2024-09-03.

6.5 Quantitative Model Comparison

The unsupervised metrics defined in Section 5.8 were computed for each model and day. Table 6.3 reports mean ± standard deviation across 74 days.

Metric	LSTM-AE	Conv-AE	VAE-Prob	RDSSM
Pct. time anomalous	1.139 ± 2.428	1.173 ± 2.348	0.946 ± 2.565	0.928 ± 0.879
Event count	11.03 ± 8.27	7.20 ± 4.92	10.01 ± 5.10	6.89 ± 5.04
Event duration median (s)	38.8 ± 28.3	65.1 ± 55.4	35.7 ± 32.4	73.7 ± 49.6
Event duration P95 (s)	103.7 ± 70.8	154.3 ± 133.0	73.8 ± 66.6	137.7 ± 111.0
Mean time between events (s)	6225.4 ± 4986.9	7352.8 ± 7411.4	5061.0 ± 2271.2	7881.7 ± 5322.7
Burstiness	1.21 ± 0.70	1.14 ± 0.87	0.85 ± 0.65	1.12 ± 0.73
Max-over-median ratio	43.27 ± 67.82	47.19 ± 49.01	12.65 ± 11.36	8.89 ± 13.05
Fragmentation	0.046 ± 0.082	0.041 ± 0.079	0.079 ± 0.125	0.055 ± 0.150

Table 6.3: Aggregated Metrics

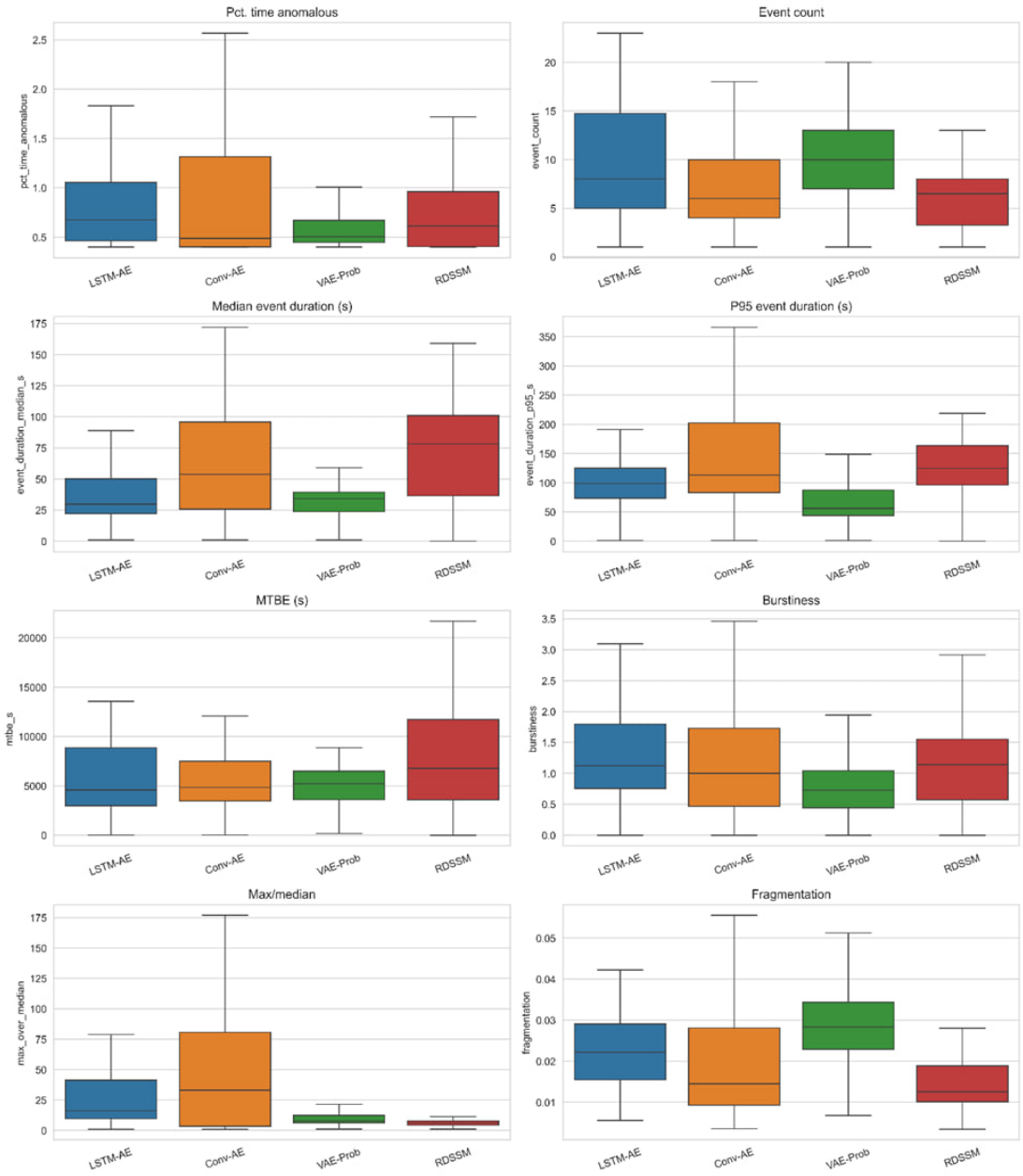


Figure 6.3: Boxplot comparison of event-level anomaly detection

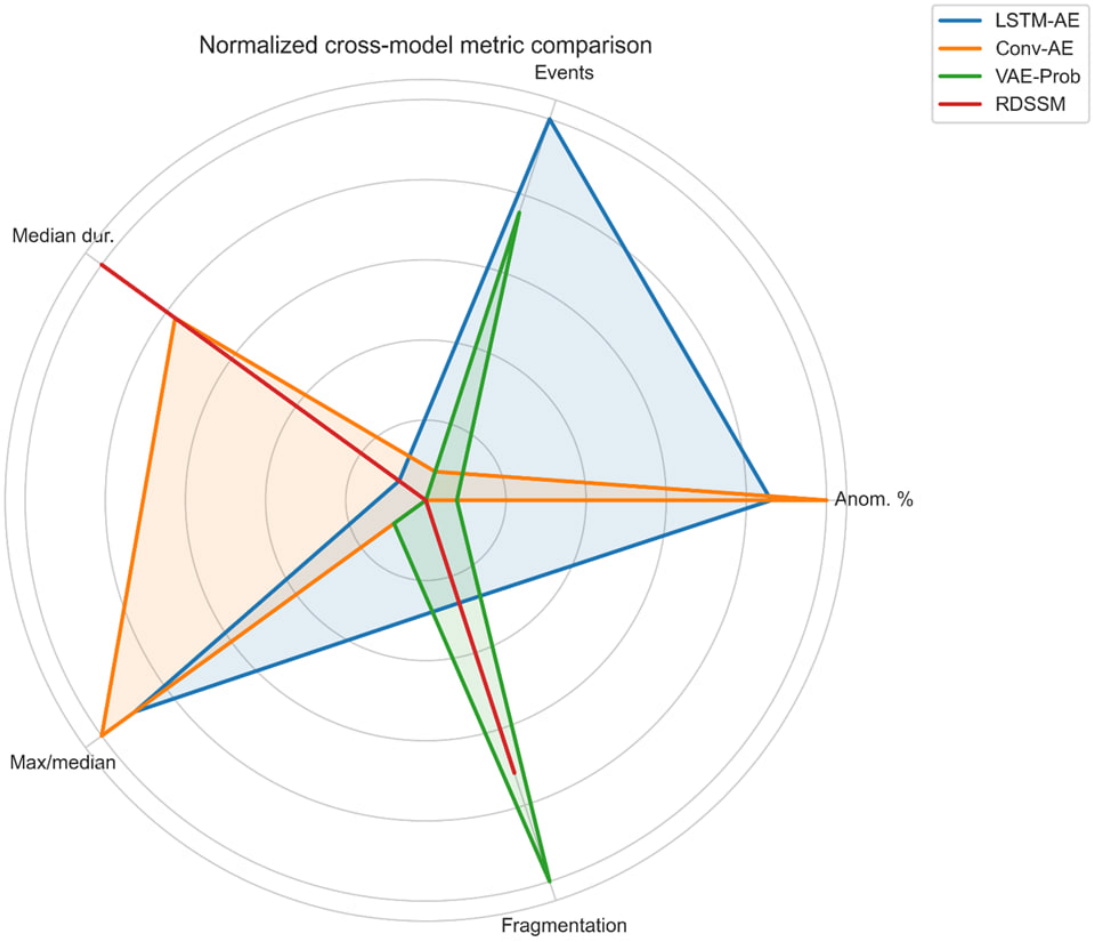


Figure 6.4: normalized cross-model metric comparison

The real-data analysis still shows no single model dominating every unsupervised metric. Instead, each model exhibits a distinctive behavioral signature in terms of event count, duration, burstiness, and fragmentation.

6.6 Event-Level Results

After aggregating anomalous indices into contiguous events as described in Chapter 5, model-specific behavioral differences become clearer.

- **LSTM-AE** produces the most frequent events on average and remains sensitive to short-lived temporal deviations.

Table 6.3 rewards a closer reading, because the aggregated metrics reveal a coherent behavioral profile for each model rather than a single winner. The LSTM-AE flags the largest share of time as anomalous (1.14% on average) and produces the most events per day (11.0), but its events are the shortest (median 38.8 s): it behaves as a *fine-grained, high-frequency sentinel*, quick to raise its hand and quick to lower it. The Conv-AE and the RDSSM occupy the opposite corner: fewer events per day (7.2 and 6.9 respectively) but substantially longer ones (median durations of 65.1 s and 73.7 s), indicating that when they fire, they tend to stay on for the whole duration of the disturbance – a *consolidated-alarm profile*. The VAE-Prob sits in between on almost every dimension (10.0 events of 35.7 s median duration), consistent with its intermediate architectural position.

Two further contrasts are informative. First, the max-over-median separability ratios of the deterministic models (43.3 and 47.2) far exceed those of the probabilistic models (12.7 and 8.9): reconstruction error explodes on rare extreme windows, while likelihood-based scores compress the extremes – again a property of the scoring formulation, not of detection quality. Second, fragmentation is lowest for Conv-AE (0.041) and highest for VAE-Prob (0.079), meaning that Conv-AE detections arrive as compact blocks whereas VAE-Prob detections are somewhat more scattered. None of these profiles is universally "better": which one is preferable depends on whether the downstream application values early fine-grained alerts or coherent, low-noise alarm segments – a point developed in Chapter 7.

- **Conv-AE** and **RDSSM** tend to produce fewer but longer events, consistent with more consolidated detections.
- **VAE-Prob** occupies an intermediate position between these two extremes.

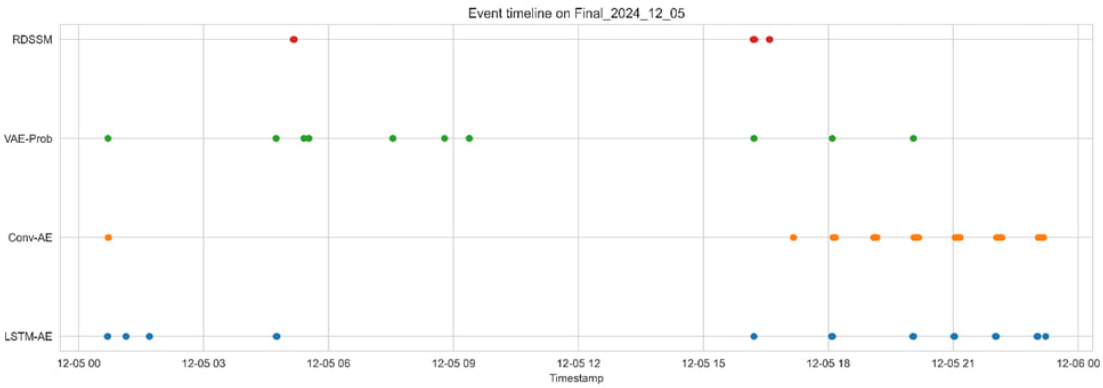


Figure 6.5: Event timeline on Final-2024-12-05

Model	Number of Events	Median Duration (s)	Max Duration (s)	Max Score in Event
LSTM-AE	12	44	105	1.847
Conv-AE	9	310	485	4.241
VAE-Prob	10	34.5	72	0.812
RDSSM	3	105	162	0.053

Table 6.4: Event-level statistics on 2024-12-05

6.7 Temporal Stability of Anomaly Scores

Visual inspection of score trajectories suggests distinct stability characteristics:

- **LSTM-AE**: produces the smoothest and most gradually varying score profiles.
- **Conv-AE**: generates sharper peaks in response to localized disruptions.

The single-day snapshot of Table 6.4 (2024-12-05) illustrates the aggregate patterns at the scale of one concrete day. The LSTM-AE reports twelve short events, the Conv-AE nine much longer ones (median 310 s, maximum 485 s), the VAE-Prob ten brief events, and the RDSSM only three, of intermediate duration. If these detections were forwarded to a maintenance operator, the practical experience would differ markedly: the LSTM-AE would deliver a stream of brief notifications, several of which likely correspond to transient and operationally harmless deviations, whereas the Conv-AE and RDSSM would deliver a handful of sustained alarms, each much more likely to warrant investigation. The event timeline in the accompanying figure also shows that the models' events cluster in the same two regions of the day (early morning and late afternoon-evening) while disagreeing on exact boundaries – an early visual indication of the partial-overlap phenomenon quantified by the Jaccard analysis of Section 6.9.

- **VAE-Prob**: exhibits moderate smoothing from latent-variable averaging.
- **RDSSM**: produces the sharpest likelihood-based deviations, often aligned with inferred state transitions.

These patterns are evident in Figure 6.1 and align with the architectural inductive biases described in Chapter 4.

6.8 Sensitivity Analysis

6.8.1 Effect of Window Size

The baseline window size is $W = 120$ s. A full retraining sweep for $W \in \{60, 240\}$ was not conducted due to the computational cost of retraining all four models. However, expected effects follow architectural properties:

- Shorter windows ($W = 60$) increase temporal resolution but reduce context, likely increasing fragmentation and reducing separability for models relying on longer temporal dependencies (LSTM-AE, RDSSM).
- Longer windows ($W = 240$) capture broader context but can smooth short anomalies, potentially reducing sensitivity to brief events—especially for locally focused architectures (Conv-AE).
- $W = 120$ provides a practical balance, capturing meaningful short-term dynamics without excessive computational burden.

6.8.2 Effect of Threshold Percentile

Threshold percentile $p \in \{99.0, 99.5, 99.9\}$ directly controls anomaly rate and event frequency. Thresholds were recomputed with no retraining, and detection statistics were recorded.

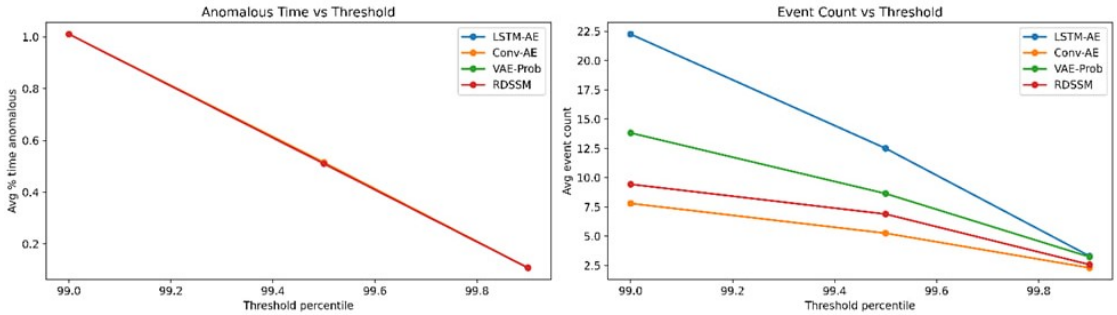


Figure 6.6: Threshold percentile sensitivity

The relative behavior of the models remains stable under threshold variation, which supports the robustness of the qualitative conclusions.

6.8.3 Effect of Latent Dimension

For VAE-Prob and RDSSM, the final reported baseline still uses the established latent-dimension configuration from the original work. A full retraining sweep across latent dimensions was not performed for the corrected 75-day rerun due to computational cost. This remains a future-work direction rather than a resolved part of the final benchmark.

6.9 Cross-Model Agreement Analysis

To assess whether different models identify the same anomalous periods, cross-model overlap was computed using the **Jaccard similarity** between sets of flagged anomaly indices [30].

$$J(A_i, A_j) = \frac{|A_i \cap A_j|}{|A_i \cup A_j|}$$

where A_i and A_j denote the sets of anomalous time indices, or time steps, flagged by models i and j.

To assess whether different models detect the same anomalous periods, a cross-model agreement analysis was conducted using the Jaccard similarity index over detected anomalous time steps [5].

On the representative day 2024-12-05, pairwise similarities are:

Model A	Model B	Jaccard
LSTM-AE	**Conv-AE**	0
LSTM-AE	**VAE-Prob**	0.115
LSTM-AE	RDSSM	0.02
Conv-AE	VAE-Prob	0
Conv-AE	RDSSM	0
VAE-Prob	RDSSM	0.079

Table 6.5: Jaccard similarity

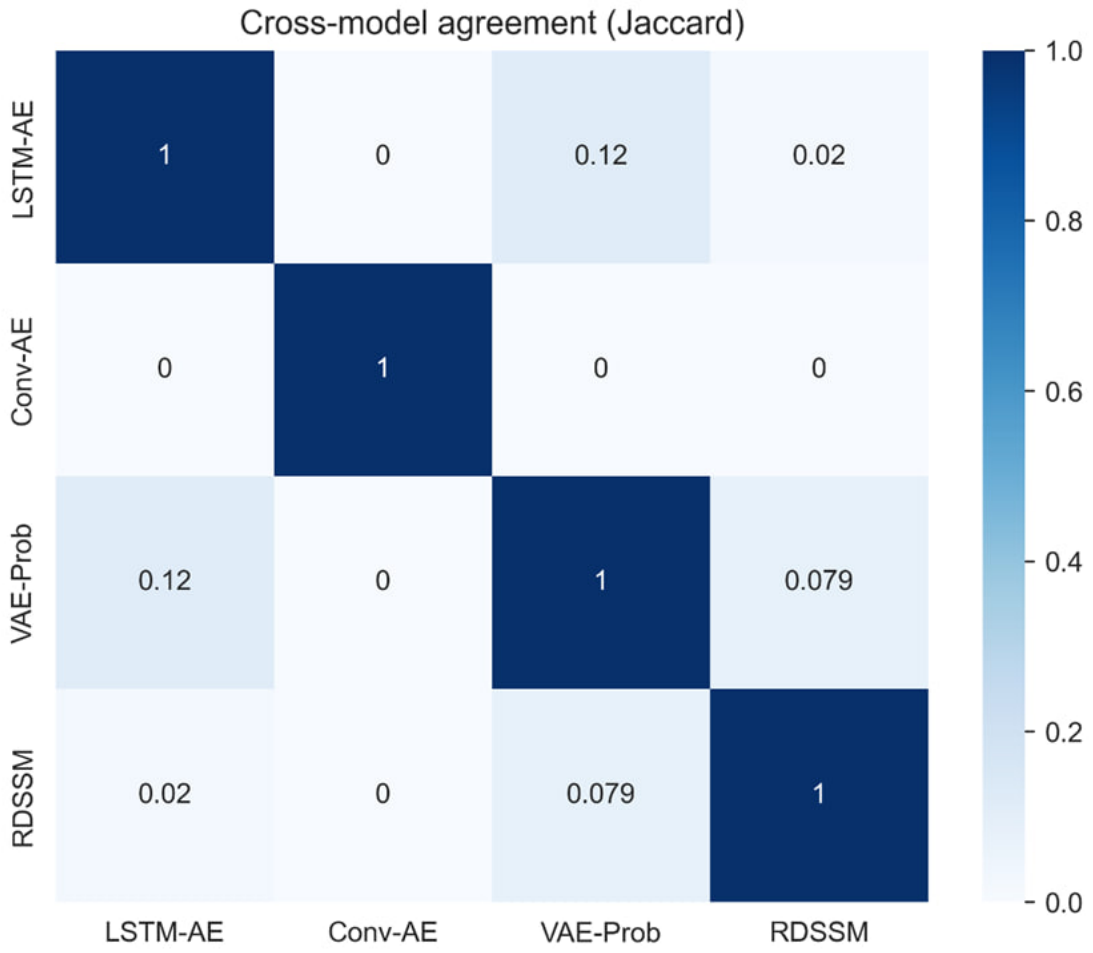


Figure 6.7: Cross-model agreement(jaccard)

The generally low Jaccard values indicate that the models remain complementary rather than redundant. This supports the interpretation that different architectures emphasize different aspects of abnormal behavior.

6.10 Synthetic Anomaly Benchmark

The synthetic anomaly benchmark introduces controlled validation-only anomalies and provides objective detection metrics while preserving unsupervised training [29].

The uniformly low Jaccard values, all pairs below 0.12, with Conv-AE sharing no flagged time steps with any other model on the representative day, admit two readings, and it is worth being explicit about why the second is the correct one. Read naively, low agreement could suggest that the models are unreliable—if they were all detecting "the truth," they should agree. Read correctly, in the light of Chapter 4, low agreement is the expected consequence of deliberately different definitions of abnormality: the Conv-AE flags broken local shapes, the LSTM-AE flags disrupted long-range order, the VAE-Prob flags statistically implausible joint configurations, and the RDSSM flags violated dynamics. These are four partially disjoint slices of the space of possible anomalies, so partial disagreement at the level of individual seconds is natural [21]. Crucially, the disagreement is not total: the highest overlaps involve the probabilistic models and the LSTM-AE, and—as shown in Section 6.6—all four models independently concentrate their strongest detections in the same early-September-2024 period. The practical implication is that the models are complementary rather than redundant: agreement across paradigms constitutes strong evidence of a genuine anomaly, while single-model detections indicate anomaly *types* that only that architecture is positioned to see.This directly motivates the ensemble direction discussed in Chapters 7 and 8.

The corrected official baseline synthetic summary below is reported on the held-out synthetic test split with **178 cases per model** after threshold calibration and minimum-duration post-processing on separate synthetic calibration days:

Model	Precision	Recall	F1	AUROC	AUPRC	Event-detection rate	Detection delay	Cases
Conv-AE	0.41	0.496	0.400	0.916	0.554	0.590	114.162	178
LSTM-AE	0.432	0.736	0.498	0.931	0.595	0.933	104.355	178
RDSSM	0.942	0.887	0.903	0.984	0.954	1.000	69.146	178
VAE-Prob	0.631	0.593	0.583	0.933	0.679	0.854	158.862	178

Table 6.6: Synthetic anomaly benchmark summary

The strongest overall baseline synthetic performer is **RDSSM**, which achieved mean precision 0.942, recall 0.887, F1 0.903, and event detection rate 1.000.

Per-scenario performance and model sensitivity are illustrated in the updated figures:

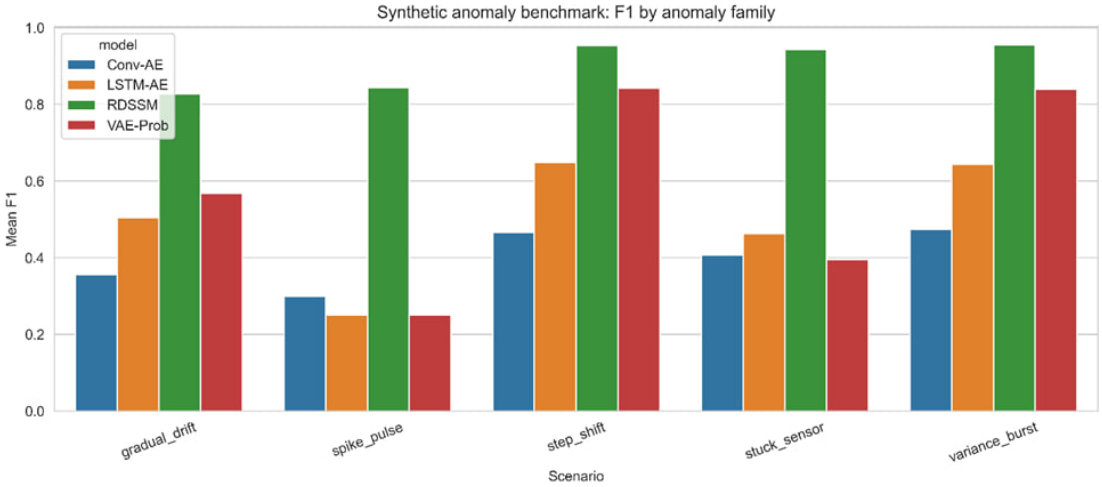


Figure 6.8: Synthetic anomaly benchmark: F1 by anomaly family

The synthetic benchmark is the point where the four behavioral profiles are finally confronted with known ground truth, and the resulting ranking is considerably sharper than anything the unsupervised metrics could establish. The RDSSM dominates on every metric: precision 0.942, recall 0.887, F1 0.903, AUROC 0.984, a perfect event detection rate, meaning every injected event triggered at least one detection, and the shortest mean detection delay (69 s). The per-scenario breakdown explains *why* it wins: it is the only model that remains strong on the spike-pulse family ($F1 \approx 0.84$ where the others fall below 0.30), because a short multivariate pulse is precisely a violation of state dynamics – visible to a transition model, but too brief to accumulate much reconstruction error.

The LSTM-AE emerges as the best deterministic baseline in terms of coverage: its recall (0.736) and event detection rate (0.933) are high, but its precision (0.432) is roughly half its recall, meaning that almost every real event is caught at the cost of many false alarms. The VAE-Prob is more balanced (precision 0.631, recall 0.593) but pays for its lack of temporal structure with the longest detection delay of all models (159 s). The Conv-AE is the weakest baseline overall (F1 0.400, event detection rate 0.590), and its per-scenario profile is exactly what its architecture predicts: acceptable on step shifts and variance bursts, which produce strong local signatures, but poor on gradual drifts, which never violate any local pattern. It is also worth noting the gap between ranking-quality and decision-quality metrics: all four models achieve AUROC above 0.91, so all four *order* windows sensibly; the large F1 differences arise at the thresholding stage, underlining once more that calibration is as important as architecture.

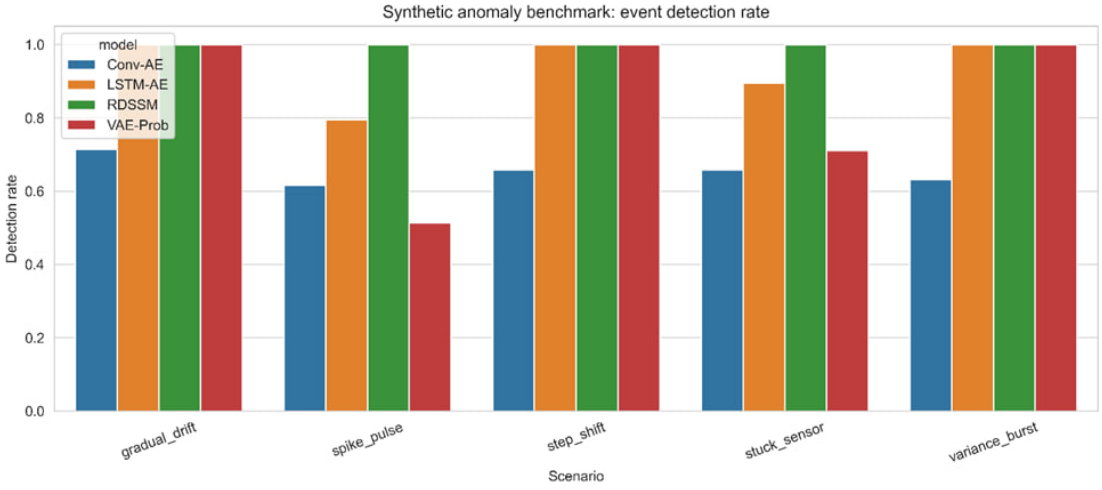


Figure 6.9: Synthetic anomaly benchmark: event detection rate

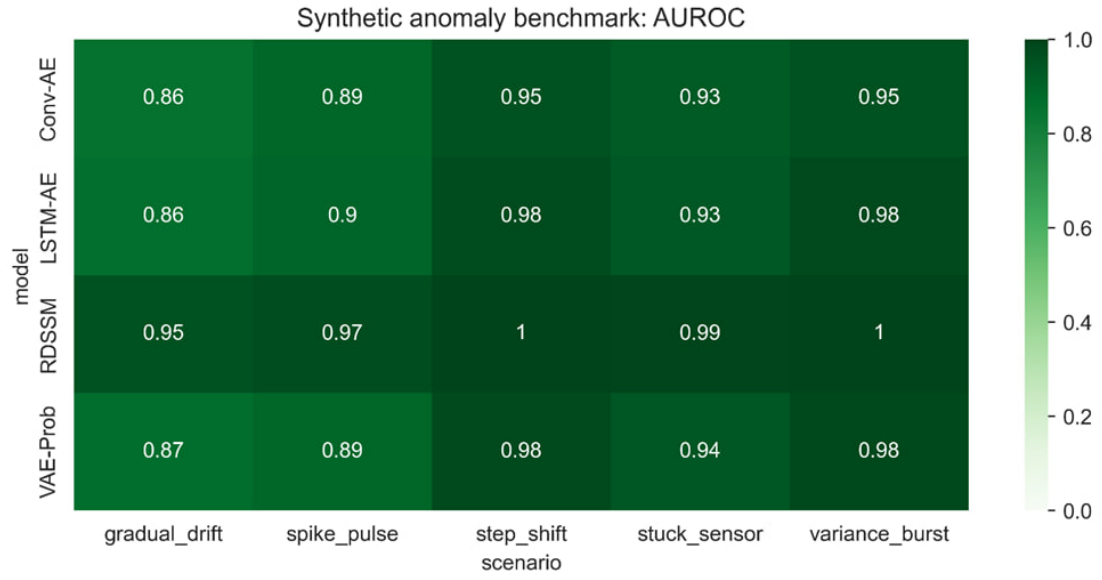


Figure 6.10: Synthetic anomaly benchmark: AUROC

6.11 Improvement Results

The improvement laboratory preserved the baseline and then fine-tuned the four models using the shared denoising strategy described in Chapter 5. The official

improved synthetic summary, again evaluated on the same held-out **178 cases per model**, is:

Model	Precision	Recall	F1	AUROC	AUPRC	Event-detection rate	Detection delay	Cases
Conv-AE	0.799	0.781	0.758	0.960	0.860	0.972	110.306	178
LSTM-AE	0.518	0.718	0.560	0.939	0.658	0.921	114.872	178
RDSSM	0.951	0.898	0.917	0.984	0.956	1.000	63.146	178
VAE-Prob	0.697	0.808	0.718	0.956	0.831	1.000	96.685	178

Table 6.7: Improved synthetic anomaly benchmark summary

The calibrated post-processing parameters used in the improvement lab were:

Model	Threshold multiplier	Min run
Conv-AE	1.2	30
LSTM-AE	0.8	30
RDSSM	3	10
VAE-Prob	1	30

Table 6.8: Calibrated post-processing parameters

In the improved study, **RDSSM** remains the strongest detector overall, while the first three models also become meaningfully stronger under the same evaluation target.

6.12 Delta Comparison Improved vs Baseline

The direct model-by-model deltas between the corrected baseline and the improvement lab are:

Model	Precision	Recall	F1	AUROC	AUPRC	Event-detection rate	Detection delay	Cases
Conv-AE	0.389	0.285	0.358	0.044	0.306	0.382	-9.556	178
LSTM-AE	0.086	-0.018	0.061	0.007	0.063	-0.011	10.517	178
RDSSM	0.009	0.011	0.015	0.001	0.001	0	-5.798	178
VAE-Prob	0.066	0.214	0.135	0.023	0.152	0.146	-62.176	178

Table 6.9: Delta comparison between improved and baseline synthetic results

The improved results reshape the comparison in an instructive way. The RDSSM improves only marginally ($F1$ from 0.903 to 0.917) – unsurprisingly, since it was already close to the ceiling – but consolidates its lead and further reduces its

detection delay to 63 s. The most dramatic change concerns the Conv-AE, whose F1 nearly doubles (0.400 → 0.758) and whose event detection rate rises from 0.590 to 0.972: denoising fine-tuning teaches it to ignore harmless local fluctuations, converting its abundant false alarms into usable precision (0.410 → 0.799). The VAE-Prob also benefits substantially ($F1$ 0.583 → 0.718) and, notably, achieves a perfect event detection rate while cutting its detection delay by more than a minute (159 s → 97 s). The LSTM-AE improves more modestly on the headline metrics ($F1$ 0.498 → 0.560) at the price of a slight recall reduction (0.736 → 0.718) and a marginally longer delay – an acceptable trade, since its main weakness was precision, not coverage. After improvement, the four models are no longer separated by an unbridgeable gap: three of them, RDSSM, Conv-AE, and VAE-Prob, reach F1 above 0.70, and the choice among them becomes a genuine engineering trade-off between accuracy, latency, and computational cost rather than a foregone conclusion.

The largest F1 gain is achieved by **Conv-AE**, which improves by **0.358**. The improvement study strengthens all four models on the headline comparison metrics *Precision*, *F1*, *AUROC*, and *AUPRC*, although not every secondary metric moves in the same direction for every model. In particular, LSTM-AE shows a small recall and delay trade-off despite improving on the main headline metrics.

6.13 Summary of Key Findings

The final experimental evaluation reveals several consistent patterns:

1. The corrected 75-day rerun preserves the broad anomaly concentration in early September 2024.
2. The four baseline models remain behaviorally distinct on real data, especially in event structure and score sharpness.
3. The synthetic benchmark provides a much clearer ranking than the unsupervised metrics alone.
4. RDSSM is the strongest standalone baseline detector on the controlled synthetic benchmark.
5. The improvement-lab study demonstrates that a shared denoising fine-tuning strategy can improve all four models on the headline comparison metrics while keeping the baseline preserved.

6. The final comparison is defendable because both baseline and improved results use the same held-out synthetic test split and the same 178-case evaluation count per model.

Chapter 7

Discussion

This chapter interprets the experimental results reported in Chapter 6 and discusses the implications of the observed differences between deterministic and probabilistic approaches to unsupervised anomaly detection. The discussion focuses on how modeling assumptions influence score behavior, temporal sensitivity, event structure, robustness considerations, and practical deployment choices.

The delta table condenses the central methodological message of the improvement study: a *single, shared, architecture-agnostic* fine-tuning strategy – denoising on normal training windows only, with the evaluation target held strictly fixed – improved every model on every headline metric: precision, F1, AUROC, and AUPRC. The magnitude of the gain, however, is inversely related to the baseline strength: the Conv-AE, weakest at baseline, gains the most (+0.358 F1); the RDSSM, strongest at baseline, gains the least (+0.015 F1). This pattern suggests that the improvement acts primarily by repairing *robustness deficits* – teaching reconstruction-based models to tolerate harmless noise – rather than by adding new detection capability, which is consistent with the fact that the models' relative per-scenario strengths and weaknesses are preserved after fine-tuning. The few secondary metrics that move against the trend, namely the LSTM-AE's small recall and delay regressions, are the expected cost of shifting a very permissive detector toward higher precision, and they do not alter the overall conclusion that the improvement is genuine, general, and honestly measured on the same held-out 178 test cases per model as the baseline.

7.1 Deterministic vs. Probabilistic Modeling Paradigms

This chapter interprets the final experimental results presented in Chapter 6, analyzing the implications of the observed differences between deterministic and probabilistic anomaly detection approaches, as well as the methodological impact of the synthetic benchmark and the improvement-lab study.

Deterministic reconstruction-based models (LSTM-AE, Conv-AE) quantify abnormality through reconstruction error, i.e., the discrepancy between observed windows and their reconstructions. As a consequence, their anomaly scores reflect *signal distortion* in a direct and intuitive way: deviations in amplitude, shape, or local pattern structure typically increase reconstruction error and therefore increase the anomaly score. This supports interpretability, since high scores can often be traced back to visibly corrupted or poorly reconstructed signal segments.

Probabilistic models (VAE-Prob, RDSSM) instead define abnormality in terms of *statistical implausibility* under a learned distribution of normal behavior. Their scoring functions are derived from likelihood-based formulations, or ELBO-based proxies, meaning that an observation can be anomalous even if its raw magnitude does not change drastically—provided it is unlikely given the learned latent distribution or latent dynamics. In this paradigm, abnormality corresponds to "surprise" relative to expected structure, rather than simple mismatch.

This conceptual distinction explains the observed differences in score scale, temporal sensitivity, and event concentration patterns, and highlights how modeling assumptions directly shape what each model considers anomalous.

7.2 Interpretation of Numerical Scale Differences

A prominent and still important finding is the numerical scale disparity observed across models. In the corrected rerun, the median clean thresholds are approximately 0.5925 for LSTM-AE, 0.9257 for Conv-AE, 0.6063 for VAE-Prob, and 0.0209 for RDSSM. This variation highlights that each model operates within its own scoring range, reflecting differences in architecture, reconstruction behavior, and probabilistic formulation.

Importantly, these scale differences do not imply superior or inferior performance. In likelihood-based frameworks, even very small absolute variations can correspond

to significant shifts in probabilistic confidence. Therefore, all thresholding and performance comparisons in this thesis are conducted in a model-relative manner, rather than relying on universal absolute values, ensuring a fair and scientifically consistent evaluation across heterogeneous model types.

7.3 Temporal Sensitivity and Pattern Capture

The final rerun reinforces the interpretation that model architecture fundamentally shapes temporal sensitivity and pattern recognition behavior. The LSTM-AE remains well suited for capturing gradual pattern distortions and smooth temporal deviations due to its sequential memory structure. In contrast, the Conv-AE demonstrates strong responsiveness to local and abrupt disturbances, as its convolutional filters emphasize short-range dependencies. The VAE-Prob model captures distributional irregularities with a moderate level of smoothing, balancing reconstruction fidelity and probabilistic representation. Meanwhile, the RDSSM exhibits heightened sensitivity to changes in latent dynamics, making it particularly effective for detecting sustained regime shifts over time.

The synthetic benchmark further supports this interpretation, as it includes a diverse set of anomaly families encompassing both abrupt and sustained perturbations. The differing responses of the models across these anomaly types highlight their inherent architectural biases toward specific temporal patterns. This behavior confirms that model performance is not solely determined by overall accuracy metrics, but also by how well each architecture aligns with the temporal structure of the anomalies being detected.

7.4 Event Fragmentation vs. Coherence

Event-level analysis continues to reveal different grouping behaviors:

- LSTM-AE tends to produce more frequent and shorter events.
- Conv-AE and RDSSM produce fewer but more consolidated detections.
- VAE-Prob occupies a middle position.

This remains practically relevant because the preferred detector depends on whether the target application values early fine-grained sensitivity or more coherent sustained alarm segments.

7.5 Cross-Model Agreement and Confidence

Cross-model agreement remains informative even though the pairwise Jaccard values are generally low. The low overlap should not be interpreted as failure; instead, it shows that the models are capturing partially different views of abnormality. This is one reason why consensus between different paradigms can be treated as stronger evidence than a single-model detection.

The corrected baseline rerun again shows that all four models concentrate their strongest detections around the early-September-2024 period, which strengthens confidence that this period contains genuinely unusual operating behavior.

7.6 Stability Under Noise and Data Quality

As documented in Chapter 3, the audited final dataset is temporally complete at 1 Hz, meaning that the final results are not driven by actual missing-value imputation artifacts. Nevertheless, the robustness question remains relevant because industrial deployments rarely stay this clean indefinitely.

The improvement results are particularly meaningful in this context. By fine-tuning the models as denoising detectors, the study explicitly pushes them to reconstruct clean normal structure from mildly corrupted normal inputs. This is a practical way to improve robustness to harmless fluctuations without changing the benchmark target.

7.7 Strengths and Limitations of Each Model

Model	Strengths	Limitations
LSTM-AE	Strong temporal modeling; stable training; interpretable scores; strong score separation	No explicit uncertainty; sensitive to scaling; higher computational cost than Conv-AE
Conv-AE	Efficient; strong local pattern detection; responsive to abrupt changes	Limited long-range modeling; may miss slow degradation
VAE-Prob	Uncertainty-aware; probabilistic interpretation; improved robustness to noise	No explicit temporal structure; more complex training; reduced interpretability vs reconstruction-only scores
RDSSM	Explicit dynamic modeling; strong for regime shifts; principled state transition reasoning	Highest complexity; sensitive to assumptions; smallest numerical score scale, requiring careful calibration

Table 7.1: Strengths and limitations of each model

7.8 Limitations of the Study

Several limitations constrain the conclusions:

1. **Lack of dense ground-truth labels** prevents supervised evaluation, including precision, recall, and F1, and limits claims about absolute accuracy.
2. **Percentile-based thresholding** enforces a fixed anomaly rate, which can obscure differences in true anomaly prevalence.
3. **Global training across days** improves generalization but does not explicitly model cross-day temporal continuity or multi-day degradation trends.
4. **Limited hyperparameter sweeps**: sensitivity analysis covers key parameters, but exhaustive optimization was not performed.
5. **Single split ratio**: only an 80/20 temporal split was used; alternative splits or cross-validation could further support robustness.

These limitations motivate future work on labeled evaluation, operational validation, and broader sensitivity analysis.

7.9 Practical Implications

Although this thesis does not implement a full predictive maintenance system, several practical implications emerge:

1. **Architecture determines detection behavior.** Model choice should be aligned with the anomaly type of interest, such as gradual degradation, abrupt faults, or regime shifts.
2. **Probabilistic scoring adds value**, especially in noisy environments, by incorporating uncertainty and distributional reasoning rather than only reconstruction mismatch.
3. **Multi-model ensembles** offer a promising path for improving coverage and confidence, especially when agreement across paradigms is used for prioritization.

4. **A synthetic-ground-truth benchmark** is highly valuable when reliable labels are unavailable.
5. Model-specific **threshold calibration** is essential.
6. **Threshold calibration is inherently model-specific** and must be defined relative to each model's score distribution.
7. **Preprocessing quality is foundational.** A consistent, structured pipeline reduces the risk that downstream differences are driven by data preparation artifacts rather than model assumptions.

7.10 Improvement Study

This thesis does not claim improvement by altering or redefining the evaluation target. Instead, it demonstrates improvement through systematic enhancements to the models themselves, while strictly keeping the evaluation framework, dataset, and benchmarking protocol fixed. By preserving a consistent target, the comparison remains fair, controlled, and scientifically valid, ensuring that any observed gains can be directly attributed to the proposed modeling improvements rather than changes in evaluation criteria.

In a label-scarce setting, where reliable ground truth is limited or partially constructed, this approach represents one of the strongest forms of empirical evidence. It eliminates ambiguity around shifting baselines and reinforces the credibility of the results, as improvements are measured against an unchanged and well-defined benchmark. This methodological rigor strengthens the overall validity of the conclusions and supports a more trustworthy comparison across models and experimental conditions.

Chapter 8

Conclusion and Future Work

8.1 Summary of the Work

This thesis addressed the problem of detecting anomalies in the traction system of an operating railway vehicle, using fourteen months of multivariate sensor data, consisting of 75 daily recordings, 88 channels, and 1 Hz sampling frequency, collected by CRRC in collaboration with BLUE Engineering, under the defining constraint of real industrial practice: the absence of reliable anomaly labels. The problem was formalized as learning, in a fully unsupervised manner, a scoring function that assigns high values to time windows inconsistent with normal system behavior, as introduced in Section 1.4, and it was attacked through a controlled comparison of four deep learning architectures spanning two paradigms: deterministic reconstruction, represented by LSTM-AE and Conv-AE, and probabilistic likelihood modeling, represented by VAE-Prob and RDSSM.

Three methodological pillars supported the study. First, a robust preprocessing pipeline enforcing a strict one-second temporal grid, structured gap detection, differentiated imputation, and sensor-wise normalization, guaranteeing that downstream differences reflect models rather than data preparation. Second, a unified experimental framework applying identical windowing, training, scoring, thresholding, and event-aggregation procedures to all four models. Third, a two-layer evaluation strategy: unsupervised behavioral metrics on the real data, complemented by a controlled synthetic-anomaly benchmark with five anomaly families injected only into held-out validation segments, providing objective precision-recall measurements without compromising the unsupervised training protocol. A final improvement study fine-tuned all four models with a shared denoising strategy against a strictly

frozen evaluation target.

8.2 Conclusions Drawn from the Results

The experimental evidence gathered in Chapter 6 supports the following conclusions.

1. **Modeling assumptions – not raw capacity – are the primary determinant of detection behavior.** The four models, trained on identical data under identical protocols, produced systematically different anomaly signatures: the LSTM-AE acted as a fine-grained sentinel, with most events and shortest durations, smoothest scores; the Conv-AE and RDSSM as consolidated-alarm detectors, with fewest and longest events; and the VAE-Prob as an intermediate case. Each profile is the direct expression of the corresponding architectural bias – sequential memory, local convolution, distributional encoding, or state-transition dynamics. For a practitioner, the implication is concrete: model selection should start from the *type* of anomaly that matters operationally, such as abrupt fault, gradual drift, or regime shift, because each architecture is structurally predisposed to a subset of these.

2. **When confronted with ground truth, explicit modeling of both time and uncertainty wins decisively.** On the synthetic benchmark, the RDSSM, the only model combining recurrent temporal structure with probabilistic state modeling, outperformed all alternatives on every metric: F1 0.903 vs. at most 0.583 at baseline; perfect event detection rate; shortest delay. It was also the only model to reliably detect short multivariate pulses, the family invisible to reconstruction error.This result elevates a conceptual argument from Chapter 2 into an empirical finding: anomalies of *dynamics* exist in real industrial data, and only dynamics-aware models can see them. The associated cost, the highest architectural complexity and a score scale requiring careful calibration, is real but, on this evidence, worth paying when detection quality is the priority.

3. **The models are complementary, and their disagreement is informative rather than pathological.** Cross-model Jaccard overlaps were uniformly low, below 0.12, yet all four models independently concentrated their strongest detections in the same early-September-2024 period. Together, these two observations show that the models capture partially disjoint facets of abnormality while agreeing on the most severe episodes. Consensus across paradigms can therefore be treated as high-confidence evidence of genuine anomaly, a principle that makes multi-model ensembles the most promising

deployment configuration, with the RDSSM as the primary detector and the reconstruction-based models as interpretable, complementary witnesses.

4. **Targeted robustness training lifts all models without moving the goalposts.** A single shared denoising fine-tuning strategy, evaluated on the same frozen 178-case held-out test set as the baseline, improved every model on every headline metric. The gains were largest exactly where the baseline weaknesses were largest: the Conv-AE's F1 nearly doubled, from 0.400 to 0.758, as its false alarms were converted into precision, and smallest for the already-strong RDSSM, from 0.903 to 0.917. This inverse relationship indicates that the dominant failure mode of the baseline detectors was not blindness to anomalies but oversensitivity to harmless fluctuation, and that this failure mode is inexpensive to repair. After improvement, three of the four models exceed F1 = 0.70, turning model choice into a legitimate trade-off between accuracy, detection latency, and computational cost.

5. **Rigorous evaluation is achievable without labels, but only through deliberate design.** Two methodological results of this thesis are as important as the model results. First, raw anomaly scores from different families are numerically incommensurable, with median thresholds spanning 0.021 to 0.926, so all valid comparison must be performed relative to each model's own score distribution and calibrated operating point. Second, unsupervised behavioral metrics alone could characterize but not rank the models; the controlled synthetic benchmark, injected only at evaluation time and calibrated and tested on disjoint partitions, was what converted behavioral description into decisive, reproducible ranking. This evaluation architecture is itself a transferable contribution for any label-scarce industrial setting.

Overall conclusion. For unsupervised anomaly detection in railway traction systems, and by reasonable extension in comparable industrial monitoring settings, the evidence of this thesis supports a clear recommendation: deploy a probabilistic state-space detector as the primary alarm source, harden all detectors with denoising fine-tuning, calibrate thresholds per model on clean historical data, and exploit cross-paradigm agreement as a confidence signal for prioritizing maintenance interventions. The early-September-2024 episode, independently flagged by all four architectures, stands as the study's strongest candidate for a genuine operational anomaly and as a natural starting point for validation against maintenance records.

8.3 Contributions

This thesis makes the following contributions:

1. **A robust preprocessing pipeline** for high-resolution industrial sensor data, including structured gap detection, differentiated imputation strategies, and consistent normalization. The pipeline is designed to generalize to other industrial time-series settings with similar data-quality risks.
2. **A unified and reproducible experimental framework** enabling fair comparison of deterministic and probabilistic anomaly detection models under identical conditions, including windowing, training, scoring, thresholding, event aggregation, and evaluation.
3. **A comprehensive behavioral analysis** of four model families using unsupervised quality metrics, including score separability, event coherence, temporal stability, threshold robustness, and cross-model agreement, offering practical guidance for model selection in label-scarce settings.
4. **Practical insights into modeling assumptions**, demonstrating that temporal modeling strategy, uncertainty representation, and latent dynamics formulation substantially determine anomaly detection characteristics.
5. A controlled **synthetic-ground-truth benchmark** for label-scarce industrial anomaly detection.
6. A shared **denoising fine-tuning strategy** that improves all four models on the headline comparison metrics.

8.4 Limitations

The results should be interpreted in light of the following limitations:

1. **The synthetic benchmark** is controlled and informative, but not identical to real faults.
2. **Percentile-based thresholding** enforces a fixed anomaly rate by design, which may not align with operational requirements and may obscure differences in true anomaly prevalence.

3. **Global training across days** improves generalization across operating conditions but does not explicitly model cross-day continuity or longer-term degradation trajectories spanning weeks or months.

4. **Scope limited to detection.** The study does not extend to fault diagnosis, root-cause identification, or maintenance decision-making.

5. **Single dataset and train type.** Generalization to other railway platforms or industrial domains requires additional validation on independent datasets.

8.5 Future Work

Several directions can extend this work and increase practical impact:

1. **Predictive maintenance**, As a possible plan for future work, the suggested system could be extended towards predictive maintenance. By collecting and analyzing historical sensor data, operational logs, and failure-related events, machine learning models could be trained to identify early signs of abnormal behavior and estimate the remaining useful life of critical components. This would allow the system not only to monitor the current condition of the asset, but also to support maintenance planning and reduce unexpected downtime.

2. **Incorporating labeled anomaly events**, even partial labels from maintenance logs, would support supervised or semi-supervised evaluation, enabling calibration and performance assessment using precision-recall analyses.

3. **Adaptive thresholding strategies** that account for operational context, non-stationarity, and time-of-day effects could provide more operationally meaningful anomaly rates than fixed percentiles.

4. **Multi-day and cross-day modeling** would allow learning longer-term dependencies and recurring patterns across days, supporting detection of gradual degradation trends developing over weeks or months.

5. **Hybrid objectives** combining reconstruction and probabilistic scoring could merge interpretability, via reconstruction error, with uncertainty-aware detection, via likelihood-based measures. For example, recurrent probabilistic models, such as LSTM-VAE variants, may capture both temporal structure and distributional deviations more effectively.

6. **Attention-based architectures**, such as Transformer-based autoencoders, could offer improved long-range dependency modeling without sequential recurrence, potentially improving detection of sparse or delayed effects.

7. **Interpretable fault diagnosis** methods linking detected anomalies to component-level causes would bridge detection to actionable maintenance decisions, improving operational usefulness.

8. **Real-world deployment evaluation** would clarify robustness under operational constraints, including latency, computational cost, scalability, and long-term drift, enabling transition from research comparison to industrial monitoring practice.

Bibliography

[1] Varun Chandola, Arindam Banerjee, and Vipin Kumar. «Anomaly Detection: A Survey». In: *ACM Computing Surveys* 41.3 (2009), pp. 1–58. DOI: 10.1145/1541880.1541882 (cit. on pp. 4, 10).

[2] Lukas Ruff, Robert Vandermeulen, Nico Goernitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Alexander Binder, Emmanuel Müller, and Marius Kloft. «Deep One-Class Classification». In: *Proceedings of the 35th International Conference on Machine Learning*. Vol. 80. Proceedings of Machine Learning Research. PMLR, 2018, pp. 4393–4402. URL: https://proceedings.mlr.press/v80/ruff18a.html (cit. on p. 4).

[3] Ane Blazquez-Garcia, Angel Conde, Usue Mori, and Jose A. Lozano. «A Review on Outlier/Anomaly Detection in Time Series Data». In: *ACM Computing Surveys* 54.3 (2021). DOI: 10.1145/3444690 (cit. on pp. 5, 11).

[4] Marco A. F. Pimentel, David A. Clifton, Lei Clifton, and Lionel Tarassenko. «A Review of Novelty Detection». In: *Signal Processing* 99 (2014), pp. 215–249. DOI: 10.1016/j.sigpro.2013.12.026 (cit. on p. 10).

[5] Sebastian Schmidl, Phillip Wenig, and Thorsten Papenbrock. «Anomaly Detection in Time Series: A Comprehensive Evaluation». In: *Proceedings of the VLDB Endowment* 15.9 (2022), pp. 1779–1797. DOI: 10.14778/3538598.3538602 (cit. on pp. 11, 16, 53, 65).

[6] Sondre Sørbø and Massimiliano Ruocco. «Navigating the Metric Maze: A Taxonomy of Evaluation Metrics for Anomaly Detection in Time Series». In: *Data Mining and Knowledge Discovery* 38 (2024), pp. 1027–1068. DOI: 10.1007/s10618-023-00988-8 (cit. on pp. 11, 16).

[7] Mayu Sakurada and Takehisa Yairi. «Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction». In: *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*. ACM, 2014, pp. 4–11. DOI: 10.1145/2689746.2689747 (cit. on p. 12).

[8] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. «LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection». In: *arXiv preprint arXiv:1607.00148* (2016). URL: https://arxiv.org/abs/1607.00148 (cit. on p. 12).

[9] Mohsin Munir, Shoaib Ahmed Siddiqui, Andreas Dengel, and Sheraz Ahmed. «DeepAnT: A Deep Learning Approach for Unsupervised Anomaly Detection in Time Series». In: *IEEE Access* 7 (2019), pp. 1991–2005. DOI: 10.1109/ACCESS.2018.2886457 (cit. on p. 12).

[10] Chuxu Zhang et al. «A Deep Neural Network for Unsupervised Anomaly Detection and Diagnosis in Multivariate Time Series Data». In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 01. 2019, pp. 1409–1416. DOI: 10.1609/aaai.v33i01.33011409 (cit. on p. 12).

[11] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A. Zuluaga. «USAD: UnSupervised Anomaly Detection on Multivariate Time Series». In: *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2020, pp. 3395–3404. DOI: 10.1145/3394486.3403392 (cit. on p. 12).

[12] Bo Zong, Qi Song, Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen. «Deep Autoencoding Gaussian Mixture Model for Unsupervised Anomaly Detection». In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=BJJLHbb0- (cit. on pp. 13, 14).

[13] Diederik P. Kingma and Max Welling. «Auto-Encoding Variational Bayes». In: *International Conference on Learning Representations*. 2014. arXiv: 1312.6114 (cit. on pp. 13, 36).

[14] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. «Robust Anomaly Detection for Multivariate Time Series through Stochastic Recurrent Neural Network». In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2019, pp. 2828–2837. DOI: 10.1145/3292500.3330672 (cit. on p. 14).

[15] Rahul Krishnan, Uri Shalit, and David Sontag. «Structured Inference Networks for Nonlinear State Space Models». In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 31. 1. 2017. DOI: 10.1609/aaai.v31i1.10779 (cit. on pp. 14, 40).

[16] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C. Courville, and Yoshua Bengio. «A Recurrent Latent Variable Model for Sequential Data». In: *Advances in Neural Information Processing Systems*. Vol. 28. 2015 (cit. on p. 14).

[17] Hang Zhao et al. «Multivariate Time-Series Anomaly Detection via Graph Attention Network». In: *Proceedings of the IEEE International Conference on Data Mining*. 2020, pp. 841–850. DOI: 10.1109/ICDM50108.2020.00093 (cit. on p. 15).

[18] Ailin Deng and Bryan Hooi. «Graph Neural Network-Based Anomaly Detection in Multivariate Time Series». In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.5 (2021), pp. 4027–4035. DOI: 10.1609/aaai.v35i5.16523 (cit. on p. 15).

[19] Jiehui Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. «Anomaly Transformer: Time Series Anomaly Detection with Association Discrepancy». In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=LzQQ89U1qm_ (cit. on p. 15).

[20] Shreshth Tuli, Giuliano Casale, and Nicholas R. Jennings. «TranAD: Deep Transformer Networks for Anomaly Detection in Multivariate Time Series Data». In: *Proceedings of the VLDB Endowment* 15.6 (2022), pp. 1201–1214. DOI: 10.14778/3514061.3514067 (cit. on p. 15).

[21] Nesime Tatbul, Tae Jun Lee, Stan Zdonik, Mejbah Alam, and Justin Gottschlich. «Precision and Recall for Time Series». In: *Advances in Neural Information Processing Systems*. Vol. 31. 2018 (cit. on pp. 15, 67).

[22] Renjie Wu and Eamonn Keogh. «Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress». In: *IEEE Transactions on Knowledge and Data Engineering* 35.3 (2023), pp. 2421–2429. DOI: 10.1109/TKDE.2021.3112126 (cit. on pp. 16, 17).

[23] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. «LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection». In: *CoRR* abs/1607.00148 (2016). arXiv: 1607.00148 (cit. on p. 33).

[24] Sepehr Maleki, Sasan Maleki, and Nicholas R. Jennings. «Unsupervised Anomaly Detection with LSTM Autoencoders Using Statistical Data-Filtering». In: *Applied Soft Computing* 108 (2021), p. 107443. DOI: 10.1016/j.asoc.2021.107443 (cit. on p. 33).

[25] Markus Thill, Wolfgang Konen, Hao Wang, and Thomas Bäck. «Temporal Convolutional Autoencoder for Unsupervised Anomaly Detection in Time Series». In: *Applied Soft Computing* 112 (2021), p. 107751. DOI: 10.1016/j.asoc.2021.107751 (cit. on p. 35).

[26] Mohsin Munir, Shoaib Ahmed Siddiqui, Andreas Dengel, and Sheraz Ahmed. «Deep Convolutional Clustering-Based Time Series Anomaly Detection». In: *Sensors* 21.17 (2021), p. 5812. DOI: 10.3390/s21175812 (cit. on p. 35).

[27] Jinwon An and Sungzoon Cho. «Variational Autoencoder Based Anomaly Detection Using Reconstruction Probability». In: *Special Lecture on IE* 2.1 (2015), pp. 1–18 (cit. on pp. 36, 37).

[28] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. «Learning Latent Dynamics for Planning from Pixels». In: *International Conference on Machine Learning*. 2019, pp. 2555–2565 (cit. on p. 40).

[29] John Paparrizos, Yuhao Kang, Paul Boniol, Ruey S. Tsay, Themis Palpanas, and Michael J. Franklin. «TSB-UAD: An End-to-End Benchmark Suite for Univariate Time-Series Anomaly Detection». In: *Proceedings of the VLDB Endowment* 15.8 (2022), pp. 1697–1711. DOI: 10.14778/3529337.3529354 (cit. on pp. 53, 66).

[30] Paul Jaccard. «The Distribution of the Flora in the Alpine Zone». In: *New Phytologist* 11.2 (1912), pp. 37–50. DOI: 10.1111/j.1469-8137.1912.tb05611.x (cit. on p. 65).