Università degli Studi di Salerno

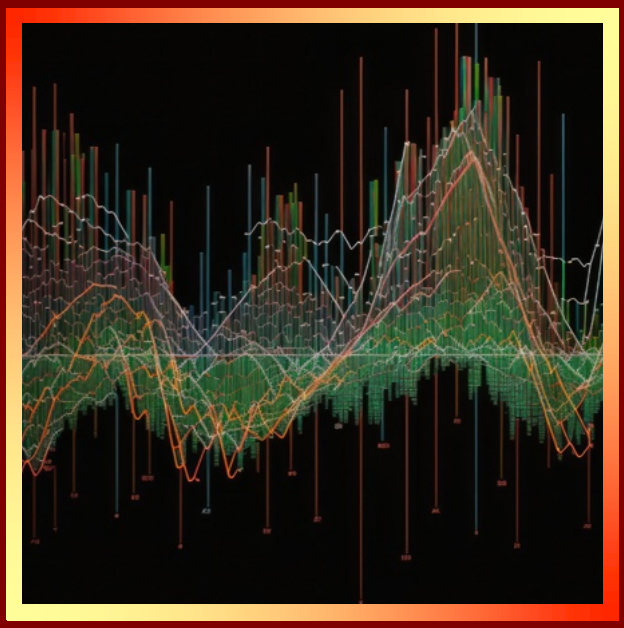
Dipartimento di Informatica

Dottorato di Ricerca in Informatica – XXXVII Ciclo

Tesi di Dottorato/PhD Thesis

Generative Models with Time Encoding for Seasonal Time Series Synthesis

Lorenzo Porcelli



Supervisor: **Prof. Francesco Palmieri**

PhD Program Director: **Prof. Andrea De Lucia**

A.A. 2024/2025

Curriculum Computer Science and Information Technology

Università degli Studi di Salerno

Dipartimento di Informatica

Dottorato di Ricerca in Informatica

Curriculum Computer Science and Information Technology

XXXVII Ciclo

TESI DI DOTTORATO / PHD THESIS

Generative Models with Time Encoding for Seasonal Time Series Synthesis

LORENZO PORCELLI

SUPERVISOR: **PROF. FRANCESCO PALMIERI**

PHD PROGRAM DIRECTOR: **PROF. ANDREA DE LUCIA**

A.A 2024/2025

This research was conducted at the University of Salerno, Department of Computer Science, under the supervision of Prof. Francesco Palmieri, from November 2021 to August 2025.

REVIEWERS:

Prof. Karl Andersson
Luleå University of Technology, Sweden

Prof. Mauro Iacono
University of Campania "L. Vanvitelli", Italy

EXAMINATION COMMITTEE:

Prof. Bruno Carpentieri
University of Salerno, Italy

Prof. Salvatore Venticinque
University of Campania "L. Vanvitelli", Italy

Prof. Michele Mastroianni
University of Foggia, Italy

First version: September 2025
Last revision: January 2026

That the powerful play goes on,
and you may contribute a verse.

ACKNOWLEDGEMENTS

I embarked on this PhD journey during the pandemic, a period of significant uncertainty. I have always viewed university as an immersive experience defined not only by study but also by daily interactions within a vibrant community. Spending much of my Master's degree remotely felt incomplete, as the university experience I loved had become impersonal.

The opportunity for a PhD arose just as the pandemic seemed to be subsiding, immediately intertwining with my professional career. Within a few weeks, I undertook both the PhD admission process and the written exam for the Bank of Italy's public competition. I learned I had been awarded the PhD scholarship even before receiving the written exam results from the Bank. A few days after the PhD program began, I took the oral exam, ranking among the successful candidates at the Bank as well. In hindsight, the choice to pursue both paths proved essential.

Navigating these two worlds offered a unique perspective that enriched me professionally and personally. The professional context allowed me to tackle real-world problems requiring concrete solutions. Academic research taught me methodological rigor and clarity of expression, enabling me to transform raw insights into shareable contributions.

Balancing a PhD and a full-time job was challenging, but through this research, I attempted to build a bridge between these two worlds. Had I followed only one path, my vision would have been narrower, leaving me unaware of this limitation. Therefore, I cannot begin this thesis without thanking those who made this unconventional journey possible and supported me over time.

I am deeply grateful to my supervisor, Prof. Francesco Palmieri, for recognizing the opportunities presented to me and supporting my choices with wisdom. He provided invaluable guidance that extended well beyond the research itself, helping me stay on track while allowing me creative freedom.

I thank Prof. Karl Andersson and Prof. Mauro Iacono for the time spent reviewing this thesis.

I would like to express my gratitude to the University of Salerno and the Bank of Italy for making this path possible. In the academic sphere, particular thanks go to the PhD Program Director, Prof. Andrea De Lucia, and the Doctoral Board for their understanding throughout these years. A special thank you also goes to Dr. Giovanni Salzano, who wisely advised me to postpone difficult decisions when the situation seemed unsustainable, allowing me to choose with a clear head.

At the Bank of Italy, I thank those who, at different times and in different roles, supported the continuation of this journey after I began my position: Silvio Orsini, Giampiero Longobardi, and Roberto Mussardo for placing their trust in me from the start, and Paolo Libri and Vincenzo Pettenello for allowing me to pursue this path in the subsequent years.

A special acknowledgment goes to my direct managers, Pietro Tiberi and Marco Capotosto, for constantly motivating me to pursue my goals. I also thank colleagues such as Alessandro Calvo, whose generosity in sharing contacts when I arrived helped me navigate the institutional environment.

I would also like to thank my coauthors Ugo Fiore, Massimo Ficco, Marcello Trovati, and Michele Mastroianni for their valuable feedback and support.

I thank my colleagues at the Bank of Italy, particularly Claudio Papa, Dario Armeni, and Mario Navarra, who provided valuable technical support within their respective areas of expertise. Despite the demands of our daily work, they were always available when I needed them. I am also grateful to all my colleagues in SPED for their collaboration, moral support, and the lighter moments that made these years more bearable.

Finally, on a personal note, I am especially grateful to my parents for the unwavering trust they have placed in me, supporting every choice I have made, and to my brother Leonardo, who was always ready to listen and discuss my ideas.

To Francesca, who has been by my side throughout this journey from the very beginning. You filled my life beyond these pages. Thank you for everything.

Fisciano, January 2026
Lorenzo

ABSTRACT

Many deep learning models for time series process input using sliding windows without including time as an explicit variable. In high-frequency contexts requiring real-time analysis, window hyperparameters demand careful calibration and errors propagate rapidly in autoregressive forecasts. In practice, models are typically supplied with a continuous stream of new observations to improve accuracy, but this renders them fragile when data are missing or delayed. This dissertation proposes an alternative approach for synthesizing seasonal time series through deep learning models that leverage geometric representations of time. The first contribution is an invertible transformation, called Helical Time Encoding (HTE), that maps time onto a higher-dimensional space where seasonal periodicity and temporal progression are geometrically embedded. The second is the Generation with the Timestamp Trick (GenTT) framework, which exploits time representations with properties analogous to HTE to train models capable of generating time series by considering only timestamps, without storing historical data or processing ordered sequences. GenTT implementations use multilayer perceptron networks, enabling deployment on resource-constrained devices without hardware acceleration and reducing energy consumption by over three orders of magnitude compared to recurrent models. We validated the GenTT framework for forecasting and anomaly detection tasks using conditional generative adversarial networks (GANs) and variational autoencoders (VAEs). In seasonal time series forecasting, GANs avoided the error propagation typical of long-term autoregressive predictions. For anomaly detection, VAEs generated expected values from learned seasonal patterns, ensuring predictions robust to both recent outliers and missing data. A variant of these models, developed for the TARGET Instant Payment Settlement (TIPS) service, enables real-time anomaly detection in high-frequency payment streams while supporting selective updates to handle model drift. Integrating this approach into a failure detection system for instant payment infrastructure facilitates explainable diagnostics that reduce analysis time and allow immediate incident response.

Keywords: Anomaly Detection, Generative Deep Learning, Instant Payment Systems, Seasonal Time Series, Time Encoding

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AE Autoencoder

AI Artificial Intelligence

AIOps Artificial Intelligence for IT Operations

ANN Artificial Neural Network

AR Autoregressive

ARIMA Autoregressive Integrated Moving Average

AUC Area Under the Curve

AUC-PR Area Under the Precision-Recall Curve

AUC-ROC Area Under the Receiver Operating Characteristic Curve

cGAN Conditional Generative Adversarial Network

CNN Convolutional Neural Network

CPU Central Processing Unit

CSM Clearing and Settlement Mechanism

cVAE Conditional Variational Autoencoder

EDR Event Detection Rate

ELBO Evidence Lower Bound

ESN Echo State Network

ETL Extract, Transform, Load

EWMA Exponentially Weighted Moving Average

EWPA Event-Wise Point Adjustment

FAR False Alarm Rate

FN False Negative

FP False Positive

FPR False Positive Rate

FTE Functional Time Encoding

GAN Generative Adversarial Network

GCN Graph Convolutional Network

GenTT Generation with the Timestamp Trick

GNN Graph Neural Network

GPU Graphics Processing Unit

GRU Gated Recurrent Unit
HTE Helical Time Encoding
ICA Independent Component Analysis
IoT Internet of Things
IT Information Technology
KL Kullback-Leibler
KPI Key Performance Indicator
LLM Large Language Model
LSTM Long Short-Term Memory
MAE Mean Absolute Error
MAPE Mean Absolute Percentage Error
MCC Matthews Correlation Coefficient
MCMC Markov chain Monte Carlo
MIDI Musical Instrument Digital Interface
MLP Multilayer Perceptron
MoE Mixture of Experts
MoGE Mixture of Generative Experts
MSE Mean Squared Error
MTS Multivariate Time Series
MTTD Mean Time to Detect
MTTR Mean Time to Respond
NAS Neural Architecture Search
NF Normalizing Flow
NLP Natural Language Processing
NSP Network Service Provider
PA Point Adjustment
PATE Proximity-Aware Time Series Anomaly Evaluation
PCA Principal Component Analysis
PR Precision-Recall
PSP Payment Service Provider
RAM Random Access Memory
RCA Root Cause Analysis
RMSE Root Mean Square Error
RNN Recurrent Neural Network

ROC Receiver Operating Characteristic
RTGS Real-Time Gross Settlement
SCT INST SEPA Instant Credit Transfer
SEK Swedish Krona
SHAP Shapley Additive Explanations
SVM Support Vector Machine
TCN Temporal Convolutional Network
TIPS TARGET Instant Payment Settlement
TN True Negative
TP True Positive
TPR True Positive Rate
UTS Univariate Time Series
VAE Variational Autoencoder
VM Virtual Machine
VUS Volume Under the Surface
VUS-PR Volume Under the Precision-Recall Surface
VUS-ROC Volume Under the Receiver Operating Characteristic Surface
XAD Explainable Anomaly Detection
XAI Explainable Artificial Intelligence

Part I

INTRODUCTION AND BACKGROUND

The longest forecast doesn't start with a single step.

1 PROBLEM STATEMENT

Modern systems generate large volumes of time series data, the analysis of which is essential for decision-making. Deep learning has demonstrated significant capabilities in handling such data. However, current approaches face practical limitations when applied to real-time scenarios involving high-frequency time series in distributed edge environments. This chapter examines these challenges and formulates the research questions guiding this dissertation.

1.1 CONTEXT AND MOTIVATION

Industries collect data to improve products, processes, and services. When ordered chronologically, such data form time series [75]. Time series data are ubiquitous across diverse domains, including finance, IT operations, healthcare, manufacturing, energy, and transportation [39, 85, 466].

Real-world time series driven by human activity often exhibit seasonal patterns reflecting behavioral regularities. High-frequency data typically display multiple overlapping cycles: hourly observations may contain daily, weekly, and annual seasonality [176]. These multi-seasonal structures appear in digital services (web traffic [139], social media [273], communications [7]), commercial operations (retail sales, call centers, ATM transactions), healthcare (hospital admissions), and utilities (electricity and water consumption) [176]. Such patterns follow user behavior cycles, with daily peaks during business hours, reduced nighttime activity, and weekly rhythms. Telemetry workloads exhibit similar dynamics, as monitoring data inherently reflect underlying activity patterns [484].

The Internet of Things (IoT), where devices communicate autonomously, has exponentially increased the volume and velocity of time series data [6], making manual analysis infeasible [16, 278].

Time series data mining automates the extraction of actionable knowledge [109, 128, 332], with forecasting and anomaly detection being key tasks across diverse domains.

Time series forecasting accounts for correlations among temporally proximate observations, seasonal variations, and underlying trends [217]. It supports decision-making by mitigating risks and enhancing operational efficiency across numerous domains, including intelligent transportation systems for demand prediction and resource allocation [381], energy consumption planning [94], and retail sales optimization [328].

Anomaly detection aims to identify unusual or unexpected patterns, although the precise formulation of this task varies across domains [85]. Applications range from healthcare, such as detecting cardiac arrests in cardiology [10], to specialized niches like detecting structural defects in jet turbines [427]. In settings where human review is feasible, anomaly detection enables domain experts to focus on relevant events rather than manually inspecting massive data volumes.

However, certain domains impose strict temporal requirements. In modern IT operations, telemetry data are collected at sub-second frequencies from distributed, virtualized infrastructures via real-time streaming [384]. The scale, velocity, and complexity of these data require automated systems capable of reacting promptly [4, 488] or preventing issues [168, 220] to avoid wasted resources and minimize troubleshooting costs.

Instant payment systems represent a particularly demanding case. The payment industry is progressively eliminating temporal barriers, extending operating hours toward 24/7/365 availability while reducing the time between transaction initiation and settlement, shifting toward real-time processing. Disruptions in such infrastructures translate directly into financial losses and reputational damage [484]. Modern systems such as TARGET Instant Payment Settlement (TIPS) [335] are designed as distributed architectures [11, 56] to meet zero-downtime service requirements. While such designs ensure scalability and resilience, they introduce additional layers of abstraction.

Managing the operational complexity of such distributed systems exceeds the capacity of traditional roles such as site reliability engineers [36] and DevOps teams. This has driven research toward artificial intelligence for IT operations (AIOps) [224], which aims to prevent issues proactively while reducing manual effort in monitoring, root cause analysis, and recovery. AIOps is a heterogeneous research field with contributions across multiple specialized domains, including anomaly detection [89, 226, 306], resource provisioning [89, 306], root cause analysis (RCA) [226], and remediation [226, 242]. The principal challenges in AIOps stem from the characteristics of operational data, which are often noisy, generated at high frequency, exhibit high dimensionality, lack labeled anomalies, and whose patterns frequently shift due to hardware, software, or platform updates [484].

In recent years, deep learning has attracted renewed attention from both academia and industry for time series tasks [180, 310, 311], as it offers distinct advantages over classical statistical [37] and machine learning methods [48, 111, 355]. Deep architectures can handle large-scale, high-frequency data, and their performance systematically improves with larger datasets [15, 190]. Moreover, they can naturally process heterogeneous and unstructured data within a unified, end-to-end differentiable framework optimized through backpropagation [270]. These properties make deep learning particularly effective in handling noisy data and variations typical of real-world environments [142, 158, 417].

1.2 RESEARCH STATEMENT

Deep learning approaches for time series predominantly process sequential data, using historical windows of observations to predict future values [213]. While effective, this paradigm often treats time implicitly as an ordering dimension rather than an explicit structural component. We focus on seasonal time series, highlighting aspects of the sequential processing paradigm that require an alternative perspective:

- **Challenges of the window-based paradigm.** Forecasting accuracy in high-frequency time series degrades as the prediction

horizon extends due to the accumulation of errors [69, 181, 470]. Multi-step forecasting approaches [129, 393, 453] attempt to mitigate this but rely on recent observations, limiting extended long-range forecasting. Moreover, the performance of these models is sensitive to windowing hyperparameters [33, 131, 152, 222, 478]. Selecting the optimal lookback window involves a complex trade-off. Windows that are too short miss long-term dependencies, while excessively long windows introduce noise and computational overhead [61, 333].

- **Limited flexibility of generative time series models.** Unlike generative models in computer vision [49, 329, 345] that synthesize images without prior specific examples, many current time series generative models [169, 252, 425, 458] rely on sequential processing. They typically inherit the sliding window constraints of discriminative architectures [26], offering limited inference flexibility despite their potential for data augmentation and uncertainty estimation [50, 422].

Moreover, deploying deep learning models in real-time distributed systems that require continuous availability, such as instant payment infrastructures, presents specific operational challenges:

- **Computational constraints at the edge.** Real-time infrastructures demand immediate decision-making. Centralized deep learning processing may introduce prohibitive latency [333], while resource-constrained edge nodes cannot execute complex models or maintain extensive historical data [180, 358]. Existing solutions often force a trade-off between the accuracy of deep models and the efficiency of lightweight statistical methods [75, 160, 180, 217].

- **Observability gaps and explainability requirements.** Current monitoring systems often operate in isolated data silos, storing metrics separately across different tools and platforms, and failing to connect technical metrics with business processes [211, 319]. This fragmentation complicates failure detection in distributed systems where dependencies extend beyond organizational boundaries [161]. Anomaly detection is increasingly

used for failure management but remains insufficient without diagnosis [311, 476]. Black-box models lacking explainability generate alert fatigue [249], while classical post-hoc explainable artificial intelligence (XAI) techniques impose computational costs incompatible with real-time requirements [137, 274, 337].

This research explores new approaches for generative deep learning models to synthesize seasonal time series through pointwise conditional generation rather than sequential processing.

To achieve this aim, the study is structured around three main objectives: (i) design a mechanism that, by exploiting invertible time encodings that explicitly represent seasonal patterns, enables generative deep learning models to synthesize seasonal time series without relying on sequential preprocessing; (ii) evaluate the effectiveness of this approach for forecasting and anomaly detection tasks, comparing performance against state-of-the-art approaches in controlled experiments; and (iii) validate the proposed approach in real-world high-frequency scenarios, conducting experiments on real-time time series of instant payment systems in resource-constrained environments.

This dissertation is framed by the following research questions:

RQ1 What geometric representations of temporal information can simultaneously encode seasonal periodicity and temporal progression while maintaining invertibility to recover the original temporal information?

RQ2 What mechanisms leverage temporal encodings to enable generative deep learning models to synthesize seasonal time series by conditioning on time representations rather than sequential historical observation windows?

RQ3 How does geometric time encoding affect prediction accuracy in deep learning architectures for long-term seasonal time series forecasting?

RQ3a To what extent does geometric time encoding influence the autoregressive multi-step-ahead forecasting performance

of fully connected, convolutional, and recurrent architectures when used as an auxiliary feature?

RQ3b How do generative adversarial networks (GANs) with geometric time encoding compare to autoregressive and rolling forecasting approaches in terms of accuracy and computational efficiency across extended horizons?

RQ4 How do variational autoencoders (VAEs) with geometric time encoding compare to sliding window prediction-based approaches in terms of detection performance and resource efficiency for anomaly detection in seasonal time series?

RQ5 How do generative models with geometric time encoding perform for streaming anomaly detection in high-frequency instant payment systems under resource constraints compared to state-of-the-art prediction-based models?

RQ6 How do generative models with geometric time encoding compare to enterprise anomaly detection solutions in terms of accuracy, timeliness, and business impact assessment for failure detection in distributed instant payment infrastructures?

Research questions RQ1 and RQ2 establish the theoretical foundations, while RQ3 through RQ6 evaluate specific instantiations of the proposed mechanism across different architectures (GANs, VAEs), tasks (forecasting, anomaly detection), and application domains (benchmark datasets, instant payment systems).

1.3 RESEARCH CONTRIBUTION

The main contributions of this thesis fall into three categories: novel theoretical approaches, experimental validation on benchmark datasets, and applied research in the instant payment domain. Table 1.1 maps each contribution to its corresponding research questions and publications.

Our primary theoretical contributions are:

- An invertible representation, called Helical Time Encoding (HTE), that explicitly captures periodicity and progression of

Thesis Part	Specific Contribution	RQ	Ref.
Novel Approaches	Formalization of HTE to explicitly represent periodicity and temporal progression in seasonal time series.	RQ1	[324]
	Design of the GenTT framework for conditional synthesis of seasonal time series without sliding-window preprocessing.	RQ2	[324]
Experimental Validation	Validation of conditional GANs with GenTT using HTE for long-term seasonal time series forecasting.	RQ3	[324]
	Validation of conditional VAEs with GenTT using HTE for prediction-based anomaly detection in seasonal time series.	RQ4	[325]
Applied Research	Evaluation of GenTT-based approach with HTE for real-time anomaly detection in high-frequency instant payment time series in resource-constrained environments.	RQ5	[322, 323]
	Formulation of a failure detection problem in distributed instant payment infrastructures and application of a MoGE solution based on GenTT with HTE.	RQ6	[322, 323]

Abbreviations: HTE: Helical Time Encoding; GenTT: Generation with the Timestamp Trick; MoGE: mixture of generative experts.

Table 1.1: Summary of thesis contributions.

seasonal time series through a geometric mapping on a helix, positioning observations in a higher-dimensional space where temporal relationships become spatial relationships.

- A generative framework called Generation with the Timestamp Trick (GenTT) that exploits explicit seasonal encodings (such as HTE) for the conditional synthesis of time series. The framework constrains generative models to output a time representation alongside target observations, leveraging this information to reconstruct the time series from independently generated single observations. This shift, from treating time as an implicit ordering dimension to encoding it as an explicit conditioning variable, enables the model to learn the joint distribution $P(\mathbf{y}, \mathbf{h})$ between observations $\mathbf{y}$ and their time representation

h. Consequently, dependency on sliding window preprocessing is eliminated, enabling history-free inference where time representations condition the latent space.

We instantiate the GenTT framework using HTE as the time representation and two generative models for different time series tasks:

- An implementation of conditional GANs based on an multilayer perceptron (MLP) architecture for long-term forecasting. By generating complete seasonal periods in a single inference step, this approach eliminates the autoregressive error accumulation typical of sequential models. Experimental results indicate that this method yields competitive or superior performance compared to established models in long-horizon scenarios where recent observations are unavailable.

- An implementation of conditional VAE for anomaly detection following the prediction-based paradigm. This method creates a stable seasonal baseline independent of recent, potentially anomalous inputs, making it robust to noise and missing data. It is efficient, reducing energy consumption and carbon emissions compared to sliding-window approaches, making it suitable for sustainable edge computing with real-time, high-frequency data.

Beyond benchmarking the proposed approaches against state-of-the-art models for forecasting and anomaly detection, we demonstrate the applicability of the GenTT framework in real-world instant payment systems. Specific contributions in this domain include:

- The extension of the GenTT framework to a mixture of experts architecture, called mixture of generative experts (MoGE), that enables selective model update mechanisms to avoid complete retraining when model drift occurs. We compared this architecture with state-of-the-art prediction-based methods for streaming anomaly detection in resource-constrained environments.

- The formulation of a failure detection problem in distributed clearing and settlement mechanism (CSM) infrastructures. We

introduced a compact representation of instant payment transactions that maps the various phases of the SEPA Instant Credit Transfer (SCT Inst) process onto distributed infrastructure components. This captures the holistic operational state of the system and enables the assessment of the business impact of incidents.

- The development of an explainable anomaly detection framework that integrates GenTT to address the failure detection problem in distributed CSM infrastructures. By enabling early detection of service degradation and localization of faults in distributed system components, this framework provides actionable insights that directly guide IT operators' remediation efforts and facilitate the assessment of the business impact of incidents.

The work presented in this thesis bridges theoretical foundations and practical implementations. Contributions span several research areas: (i) temporal encoding techniques for time series, (ii) generative models for time series synthesis, (iii) time series forecasting and anomaly detection using deep learning, (iv) artificial intelligence for IT operations (AIOps) in financial infrastructures, and (v) explainable artificial intelligence (XAI) approaches for anomaly detection.

1.4 STRUCTURE OF THE THESIS

Chapter 2 defines basic concepts and notation, while Chapter 3 reviews related work. The thesis is organized into three main parts, as shown in Table 1.1, with each part comprising chapters that detail individual contributions.

Part ii groups the novel approaches devised to address the first two research questions:

- Chapter 4 formalizes the HTE, defining its properties and its integration into deep learning architectures.

- Chapter 5 presents the GenTT framework, which leverages seasonal time encoding and generative models to offer an alter-

native to sequential processing through conditional generation, eliminating the need for sliding window preprocessing.

Part iii groups the chapters related to the experimental validation of two generative models instantiating the GenTT framework with HTE for forecasting and anomaly detection tasks:

- Chapter 6 evaluates the impact of using HTE in deep learning architectures for forecasting and introduces the HTE-cGANs model. HTE-cGANs is compared against state-of-the-art baselines in two scenarios: long-term forecasting without recent data availability, and sliding-window forecasting with recent data availability.

- Chapter 7 presents the RTAD-cVAE model for prediction-based anomaly detection. It evaluates the model's performance in both semi-supervised and unsupervised settings against statistical, machine learning, and deep learning baselines, testing robustness to missing data scenarios and evaluating energy efficiency compared to sliding-window approaches.

Part iv demonstrates the operational viability of the proposed approach within a real-world instant payment system:

- Chapter 8 instantiates the GenTT framework with HTE within a mixture of experts (MoE) architecture tailored for streaming anomaly detection of high-frequency time series data in resource-constrained environments. This architecture enables selective model retraining to manage concept drift.

- Chapter 9 formulates the failure detection problem in distributed CSM infrastructures. It introduces an explainable anomaly detection framework that leverages domain-specific feature engineering to bridge technical metrics with business processes, and proposes a concrete implementation based on the GenTT framework to address failure detection in real-time.

The thesis concludes with Chapter 10, which summarizes contributions, limitations, and real-world impact, and Chapter 11, which outlines theoretical extensions and applications beyond seasonal forecasting and anomaly detection.

2 BACKGROUND

This chapter establishes the theoretical foundations for this dissertation. We begin by introducing time series and defining anomalies in temporal data, then present the generative models used in our approach. Finally, we describe the metrics used to assess forecasting and anomaly detection performance, emphasizing series-aware metrics that account for the temporal structure of anomalies.

2.1 TIME SERIES

A time series is a sequential collection of data points indexed in temporal order [151]. Time series data capture the evolution of phenomena over time and are used in diverse domains, including financial markets, environmental monitoring, and system performance tracking.

Time series are classified as univariate or multivariate based on the number of variables observed at each time point.

A univariate time series (UTS) tracks a single variable over time. For example, recording hourly temperature levels constitutes a UTS. Formally, a UTS X with t timestamps is represented as:

$$X = (x_1, x_2, \ldots, x_t) \tag{2.1}$$

where x_i denotes the observation at timestamp $i \in T = \{1, 2, \ldots, t\}$.

A multivariate time series (MTS) simultaneously tracks multiple interdependent variables. Each variable exhibits temporal dependencies (influenced by its own past values) and spatial dependencies (influenced by other variables) [248]. For instance, simultaneously recording hourly temperature, humidity, and air pressure constitutes an MTS. Formally, an MTS X with d dimensions is represented as:

$$X = (X_1, X_2, \ldots, X_t) \tag{2.2}$$

where $X_i = (x_i^1, x_i^2, \ldots, x_i^d)$, and x_i^j denotes the observation of the j-th variable at timestamp i.

Time series can be decomposed into components that capture distinct behavioral patterns [81]:

- Trend: represents the long-term directional movement, capturing sustained growth, decline, or stability. Trends may be linear or nonlinear; for instance, population dynamics often follow complex patterns driven by demographic and socioeconomic factors.
- Seasonality: captures regular, periodic fluctuations occurring at fixed intervals such as daily, weekly, or yearly cycles. Energy consumption, for example, exhibits seasonal patterns influenced by weather conditions and geographic location.
- Cyclical variations: describe medium to long-term oscillations with variable duration and amplitude that differ from the fixed periodicity of seasonality. Economic cycles, including expansions and contractions, exemplify this component.
- Remainder: consists of irregular, unpredictable fluctuations not attributable to other components, including random noise and anomalous events such as system failures or natural disasters.

2.2 TIME SERIES FORECASTING

Time series forecasting aims to predict future values of a target variable $y_{i,t}$ for entity i at time t. A standard one-step-ahead forecasting model takes the form:

$$\hat{y}_{t+1} = f(y_{t-k:t}, \mathbf{x}_{t-k:t}, \mathbf{s}), \tag{2.3}$$

where $\hat{y}_{t+1}$ denotes the forecast, $y_{t-k:t}$ represents historical target observations over a lookback window k, $\mathbf{x}_{t-k:t}$ contains exogenous inputs, $\mathbf{s}$ captures static metadata, and $f(\cdot)$ is the learned prediction function. While this formulation focuses on univariate forecasting, it extends naturally to multivariate cases [241, 348, 357].

Many applications require predictions across multiple future time steps rather than just one-step-ahead, enabling decision-makers to visualize trends and optimize actions over an entire horizon. Multi-horizon forecasting extends the one-step-ahead formulation:

$$\hat{y}_{t+\tau} = f(y_{t-k:t}, \mathbf{x}_{t-k:t}, \mathbf{u}_{t-k:t+\tau}, \mathbf{s}, \tau), \tag{2.4}$$

where $\tau \in \{1, \ldots, \tau_{\max}\}$ is the forecast horizon, $\mathbf{u}_t$ represents known future inputs (e.g., calendar features), and $\mathbf{x}_t$ denotes inputs observable only historically. Multi-horizon forecasting approaches can be categorized as iterative or direct methods [285, 391].

Iterative (or recursive) approaches use autoregressive (AR) architectures [238, 349] that recursively feed predicted values into future time steps. The model generates multiple trajectories through repeated sampling, with final forecasts obtained by averaging. While straightforward to implement (training is identical to one-step-ahead models), iterative methods accumulate errors over longer horizons and assume that all inputs except the target are known at inference time.

Direct methods address the shortcomings of iterative approaches by producing all horizon forecasts in a single pass and leveraging both observed inputs and known future inputs. These methods produce forecasts for all horizons simultaneously using sequence-to-sequence architectures [254]. An encoder summarizes past information, while a decoder combines this summary with known future inputs (e.g., calendar features) to generate predictions. This approach uses all available information directly, avoiding error accumulation, but requires specifying a maximum forecast horizon at training time.

In operational settings, forecasting models are typically deployed using a sliding window approach. At each time step t, the model uses a fixed-size window of recent observations $y_{t-k:t}$ to generate predictions for one or multiple future horizons. As new observations become available, the window slides forward: the oldest observation is removed, the newest observation y_{t+1} is added, and the model produces updated forecasts using the refreshed window $y_{t-k+1:t+1}$. For instance, at time t, a model might predict $[\hat{y}_{t+1}, \hat{y}_{t+2}, \hat{y}_{t+3}]$ using observations $y_{t-k:t}$; then at time $t+1$, it predicts $[\hat{y}_{t+2}, \hat{y}_{t+3}, \hat{y}_{t+4}]$ using the updated window $y_{t-k+1:t+1}$ that incorporates the newly

observed y_{t+1}. Unlike iterative methods, this approach always uses actual observed values rather than previous predictions as inputs, preventing error accumulation across successive forecasting steps.

2.3 ANOMALY DETECTION IN TIME SERIES

In the anomaly detection literature, the term "anomaly" is often used interchangeably with "outlier" and related terms that vary depending on the domain and application. Although there is no universal consensus on the definition of an anomaly [57], a classical definition for outliers is provided by Hawkins [155], who describes outliers as observations that deviate so much from others as to suggest that they were generated by a different mechanism.

To avoid terminological ambiguity, we adopt a practical definition from the analyst's perspective. We associate outliers with noise, i.e., erroneous or undesired data that are not inherently interesting and can therefore be eliminated or corrected to improve data quality [341, 398]. In contrast, we define anomalies as unusual but interesting phenomena that we want to detect and analyze [3]. When such phenomena become a "new normal" that we incorporate into the model after detection, we call these observations novelties [38, 342, 394].

In the specific context of time series data, an anomaly is defined as a data point at a specific time step that exhibits unexpected behavior significantly different from previous time steps. A widely adopted classification distinguishes among point, contextual, and collective anomalies [60], though definitions may vary across different application domains. In time series:

- A point anomaly refers to a data point that deviates significantly from the overall distribution. For example, an extreme fluctuation in stock prices exemplifies a point anomaly.
- A contextual anomaly, also referred to as a conditional anomaly [380], occurs when a data point is anomalous within a specific temporal context, even though it might appear normal in the overall dataset. This type of anomaly is common in time

series. For instance, in traffic monitoring, elevated traffic between 3:00 and 4:00 constitutes a contextual anomaly, whereas the same traffic volume would be normal between 12:00 and 13:00.

- A collective anomaly represents a set of data points that collectively exhibit abnormal behavior or trend over time, even though individual data points might not be anomalous in isolation. This type of anomaly can often be identified only by analyzing the entire data sequence. An example is a marked increase in heart rate after learning the normal pattern.

Anomaly detection is the research area that studies the identification of anomalous observations through data-driven methods, models, and algorithms [342]. Based on the learning strategy, time series anomaly detection methods can be classified as supervised, semi-supervised, unsupervised, and self-supervised [180]:

- Supervised learning assumes that the training dataset contains instances labeled as normal or anomalous. This approach is rarely applicable in practice because anomalies are often unknown a priori.
- Semi-supervised learning assumes that the training dataset contains labeled instances of only one class (either normal or anomalous).
- Unsupervised learning requires no labeled data; instead, it relies on inherent data characteristics to identify anomalies.
- Self-supervised learning trains models using pseudo-labels derived from unlabeled data itself, by predicting unobserved parts from observed ones, then applies the learned representations to downstream tasks.

Regardless of the training strategy, the output of an anomaly detection model can be either an anomaly score or a label [85]. An anomaly score quantifies the deviation of an observation from its expected value, with computation specific to each model. Labels, conversely,

provide discrete classification of observations as normal or anomalous. Labels are often used for immediate reporting in monitored systems and can be generated either directly by a binary classification algorithm or derived by applying a threshold to a score produced by a detection algorithm; observations exceeding the specified threshold are deemed anomalous. In scenarios requiring differentiated actions or notifications for various anomaly conditions, labels may also comprise multiple classes.

2.4 GENERATIVE MODELS FOR TIME SERIES DATA MINING

A generative model learns the underlying distribution of data to generate new plausible examples. In time series applications, generative models serve multiple purposes: augmenting limited datasets with synthetic samples, imputing missing or corrupted data, denoising signals, and enabling privacy-preserving data synthesis through differential privacy mechanisms.

Generative models for time series data mining leverage diverse architectures, including VAEs [202], GANs [141], diffusion-based models [163], and large language models (LLMs) [52, 412]. This dissertation focuses on VAEs and GANs for time series forecasting and anomaly detection tasks. The following subsections describe the standard architectures of these models.

2.4.1 *Variational Autoencoders*

An autoencoder (AE) is an unsupervised neural network architecture designed for dimensionality reduction. It consists of an encoder and a decoder that jointly learn an encoding-decoding scheme minimizing reconstruction error:

$$(e^*, d^*) = \underset{(e,d)\in E\times D}{\arg\min}\, \mathbb{E}_{x\sim p_{\text{data}}(x)}\left[L\left(x, d(e(x))\right)\right], \quad (2.5)$$

where E and D denote the families of encoders and decoders, and L measures the reconstruction error between the input x and its reconstruction $d(e(x))$.

While a traditional AE could generate new data by sampling from its latent space and decoding, this approach is hindered by an irregular latent space structure, often due to overfitting. Reliable data generation requires the latent space to exhibit continuity and completeness: nearby latent points should decode to similar data, and samples drawn from the latent distribution should yield meaningful outputs.

A variational autoencoder (VAE) [201] addresses these limitations by regularizing the latent space through probabilistic modeling. The encoder maps inputs to the parameters of a variational distribution $q_\phi(z|x)$ (typically a diagonal Gaussian) in a latent space, while the decoder samples from this distribution to reconstruct the input. Assuming latent variables z follow a fixed prior distribution $p(z)$ (typically $\mathcal{N}(0, I)$), the generative process is expressed as:

$$p_\theta(x) = \int p_\theta(x|z)\, p(z)\, dz. \tag{2.6}$$

However, marginalizing over z is generally intractable for complex decoders $p_\theta(x|z)$. VAEs introduce a tractable solution by approximating the intractable posterior $p_\theta(z|x)$ with the variational distribution $q_\phi(z|x)$ through learned approximate inference, while the decoder models the likelihood $p_\theta(x|z)$.

The training objective maximizes the evidence lower bound (ELBO):

$$\mathcal{L}_{\text{VAE}}(\phi, \theta; x) = \mathbb{E}_{q_\phi(z|x)}\big[\log p_\theta(x|z)\big] - \text{KL}\big(q_\phi(z|x) \,\|\, p(z)\big), \tag{2.7}$$

where the first term encourages accurate reconstruction, and the second term regularizes the latent space by aligning the approximate posterior with the prior distribution. This formulation provides a principled framework for uncertainty quantification. Maximizing $\mathcal{L}_{\text{VAE}}$ with respect to (ϕ, θ) is equivalent to minimizing the Kullback-Leibler (KL) divergence between the approximate posterior and the true posterior $p_\theta(z|x)$, since $\log p_\theta(x) = \mathcal{L}_{\text{VAE}} + \text{KL}\big(q_\phi(z|x) \,\|\, p_\theta(z|x)\big)$.

In practice, the expectation over $q_\phi(z|x)$ is made differentiable through the reparameterization trick, expressing z as:

$$z = \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{2.8}$$

allowing gradients to propagate through stochastic nodes during training. For discrete data, the reconstruction term is typically implemented as binary cross-entropy loss; for continuous data, mean squared error is commonly used (assuming a Gaussian likelihood with fixed variance). Stochastic gradient ascent methods (e.g., Adam) are employed to maximize the ELBO.

The conditional variational autoencoder (cVAE) extends the VAE framework by conditioning both the encoder and decoder on additional information c, such as class labels, attributes, or other contextual variables. This conditioning enables controlled generation and reconstruction guided by specific conditions. The encoder maps input data to a conditional latent distribution $q_\phi(z|x,c)$, while the decoder reconstructs the input from a sampled latent vector and the conditioning information $p_\theta(x|z,c)$.

The training objective for cVAEs follows the same principle as standard VAEs, maximizing the conditional ELBO:

$$\mathcal{L}_{\text{cVAE}}(\phi,\theta;x,c) = \mathbb{E}_{q_\phi(z|x,c)}\big[\log p_\theta(x|z,c)\big] - \mathrm{KL}\big(q_\phi(z|x,c) \,\|\, p(z)\big). \quad (2.9)$$

In many implementations, a hyperparameter β is introduced to weight the KL divergence term (as in β-VAEs), balancing reconstruction accuracy with latent space regularization. This hyperparameter can be tuned to prioritize either reconstruction fidelity or adherence to the prior distribution, depending on application requirements. The reparameterization trick is similarly applied to enable end-to-end training through gradient-based optimization.

2.4.2 *Generative Adversarial Networks*

GANs [141] employ an adversarial framework comprising two competing neural networks: a generator $\mathcal{G}$ that creates synthetic data, and a discriminator $\mathcal{D}$ that distinguishes between real and fake samples. During training, the generator progressively improves its ability to produce realistic samples, while the discriminator becomes increasingly skilled at identifying fakes. This adversarial process continues

until the discriminator can no longer reliably distinguish generated samples from real data.

The training objective is theoretically formulated as a minimax game:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} V(\mathcal{D}, \mathcal{G}) = \mathbb{E}_{x \sim \mathbb{P}_{\mathbf{X}}} [\log \mathcal{D}(x)] + \mathbb{E}_{z \sim \mathbb{P}_{\mathbf{Z}}} [\log (1 - \mathcal{D}(\mathcal{G}(z)))], \quad (2.10)$$

where $\mathbb{P}_{\mathbf{X}}$ is the distribution over the real data domain $\mathbf{X}$, and $\mathbb{P}_{\mathbf{Z}}$ is the distribution over the latent domain $\mathbf{Z}$.

In practice, to address vanishing gradient issues during early training phases, the generator loss is often reformulated to maximize $\log \mathcal{D}(\mathcal{G}(z))$ (non-saturating loss) rather than minimizing $\log(1 - \mathcal{D}(\mathcal{G}(z)))$. Consequently, the discriminator loss $\mathcal{L}_{\mathcal{D}}$ and the practical generator loss $\mathcal{L}_{\mathcal{G}}$ are formulated as:

$$\mathcal{L}_{\mathcal{D}} = -\frac{1}{m} \sum_{i=1}^{m} \left[\log \left(\mathcal{D}(x^{(i)}) \right) + \log \left(1 - \mathcal{D} \left(\mathcal{G}(z^{(i)}) \right) \right) \right], \quad (2.11)$$

$$\mathcal{L}_{\mathcal{G}} = -\frac{1}{m} \sum_{i=1}^{m} \log \left(\mathcal{D} \left(\mathcal{G}(z^{(i)}) \right) \right), \quad (2.12)$$

where m denotes the batch size.

GANs have demonstrated remarkable success in generating high-quality data across diverse domains, including image synthesis, style transfer, and image completion. In time series applications, GANs have been successfully employed for audio generation, sequence forecasting, and imputation. However, GANs present three main training challenges: non-convergence, vanishing gradients that prevent generator learning, and mode collapse, where the generator produces samples with limited diversity, focusing on a specific subset of the data distribution rather than capturing its full complexity. Several variants of the original GAN architecture have been proposed to address these issues [145, 401]. Conversely, VAEs offer more stable training dynamics, though their approximation mechanism may compromise output quality relative to GANs.

The conditional generative adversarial network (cGAN) extends the GAN framework by conditioning both the generator and discriminator on additional information c, such as class labels, temporal information, or other contextual variables. This conditioning enables controlled generation of synthetic data that adheres to specific constraints or attributes. The generator takes both a noise vector $z \sim \mathbb{P}_{\mathbf{Z}}$ (typically $\mathcal{N}(0, I)$) and the conditioning information c as inputs to produce synthetic samples, while the discriminator evaluates whether a given sample is real or generated, also conditioned on c.

The training objective for cGANs extends the minimax formulation to incorporate conditioning. Given pairs of real data and conditions (x, c) sampled from the data distribution $\mathbb{P}_{\text{data}}$, the objective becomes:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} V(\mathcal{D}, \mathcal{G}) = \mathbb{E}_{(x,c)\sim\mathbb{P}_{\text{data}}} \left[\log \mathcal{D}(x|c)\right] + \mathbb{E}_{\substack{z\sim\mathbb{P}_{\mathbf{Z}} \\ c\sim\mathbb{P}_c}} \left[\log\left(1 - \mathcal{D}(\mathcal{G}(z|c)|c)\right)\right]. \quad (2.13)$$

This conditional formulation allows cGANs to generate diverse outputs aligned with specific input conditions, making them particularly effective for tasks requiring fine-grained control over the generated data characteristics.

2.5 EVALUATION METRICS

Evaluating model performance requires metrics tailored to the specific characteristics of the task. For forecasting, we present standard metrics that quantify prediction accuracy. For anomaly detection, we distinguish between point-based metrics, which evaluate individual time points, and series-aware metrics that account for the temporal continuity of anomalous events. The latter are particularly important when anomalies span multiple consecutive time steps, as accurate detection of event onset and extent is critical for effective real-world monitoring systems.

2.5.1 *Forecasting Metrics*

Forecasting performance is typically evaluated using metrics that quantify the discrepancy between predicted and actual values. We present the most widely adopted metrics in time series forecasting literature: mean absolute error (MAE), mean squared error (MSE), root mean square error (RMSE), and mean absolute percentage error (MAPE).

Let $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ denote the actual time series values and $\hat{\mathbf{y}} = (\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n)$ denote the predicted values over n time points.

MAE measures the average absolute deviation between predictions and actual values:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|. \tag{2.14}$$

MAE provides an intuitive measure of average prediction accuracy, with all errors weighted equally regardless of magnitude.

MSE quantifies the average squared deviation between predictions and actual values:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2. \tag{2.15}$$

MSE penalizes larger errors more heavily than MAE through quadratic weighting, making it sensitive to outliers while remaining differentiable.

RMSE, the square root of MSE, restores the error to the original data units:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2}. \tag{2.16}$$

MAPE expresses prediction error as a percentage of actual values:

$$\text{MAPE} = \frac{100}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|. \tag{2.17}$$

MAPE enables scale-independent comparison across datasets with different magnitudes. However, it is undefined when $y_i = 0$ and can be biased toward underestimation when actual values are large.

2.5.2 *Anomaly Detection Metrics*

Point-based metrics are widely used in anomaly detection, but due to the inherent continuity and temporal dependencies of time series data, they may not adequately capture performance when anomalies span multiple time steps. In real-world applications, accurately detecting the onset of an anomaly event is essential for timely alerting. Consequently, researchers have proposed series-aware metrics to better evaluate detection performance. In this section, we present the main evaluation metrics for time series anomaly detection.

2.5.2.1 *Point-based Metrics*

Point-based metrics, originally developed for information retrieval and binary classification, evaluate anomaly detection performance at individual time points. These metrics include Precision, Recall, F1-score, and Matthews correlation coefficient (MCC), which are computed based on the elements of the confusion matrix [385]:

- true positive (TP): anomaly points correctly identified as anomalies.
- true negative (TN): normal points correctly identified as normal.
- false positive (FP): normal points incorrectly identified as anomalies.
- false negative (FN): anomaly points incorrectly identified as normal.

Precision quantifies the accuracy of anomaly predictions by measuring the proportion of correctly identified anomalies among all points flagged as anomalous:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{2.18}$$

High precision indicates few false alarms, which is important in applications where false positives are costly or disruptive.

Recall measures the model's ability to detect actual anomalies by calculating the proportion of true anomalies that are successfully identified:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{2.19}$$

High recall ensures that most anomalies are detected, which is essential when missing anomalies could have severe consequences.

F1-score provides a balanced assessment by computing the harmonic mean of Precision and Recall:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{2.20}$$

This metric is particularly useful when both false positives and false negatives must be minimized simultaneously.

The Matthews correlation coefficient (MCC) provides a comprehensive measure of classification quality that accounts for all four confusion matrix elements:

$$\text{MCC} = \frac{\text{TP} \cdot \text{TN} - \text{FP} \cdot \text{FN}}{\sqrt{(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})}}. \tag{2.21}$$

MCC ranges from -1 to +1, where +1 indicates perfect prediction, 0 represents random performance, and -1 suggests complete disagreement between predictions and ground truth. Unlike F1-score, MCC incorporates true negatives, making it particularly reliable for imbalanced datasets where the majority class (normal instances) significantly outnumbers anomalies.

However, when anomalies span multiple consecutive time steps, these point-based metrics may inadequately capture the model's ability to detect entire anomalous events [174].

2.5.2.2 *Point Adjustment*

To better reflect the continuous nature of real-world anomalies, point adjustment (PA) treats an entire anomalous event as correctly detected if at least one point within the event is flagged as anomalous [445]. Using PA makes the evaluation more event-oriented, effectively increasing recall for anomalous events. However, PA can be overly lenient, potentially overestimating model performance [415].

Event-wise point adjustment (EWPA) [172] refines the evaluation by treating each anomalous segment as a single, indivisible entity. Rather than evaluating individual points, this method considers entire events: each anomalous segment contributes only once to the TP or FN count, regardless of its duration. This prevents disproportionate weighting of longer anomalous events in the overall evaluation.

Reduced-length PA [367] further refines EWPA evaluation by incorporating a severity coefficient based on anomaly duration. By penalizing or rewarding detections based on event duration, this approach mitigates bias in evaluating segments of varying lengths and provides a more nuanced performance assessment.

2.5.2.3 *Sequential Adaptability Metrics*

PA-based variants improve upon pointwise metrics but may not fully capture temporal characteristics such as early or delayed detections. Sequential adaptability metrics [136, 174, 395] evaluate the temporal alignment between predicted and actual anomalies. We present two prominent approaches: range-based metrics and proximity-aware time series anomaly evaluation (PATE).

Range-based Metrics

Real-world anomalies often manifest as continuous intervals rather than isolated points. Range-based metrics [395] evaluate detection performance over these intervals by measuring both detection occurrence and coverage quality.

Given a set of true anomalous intervals $R = \{R_1, \ldots, R_{N_r}\}$ and predicted intervals $P = \{P_1, \ldots, P_{N_p}\}$, range-based recall ($\text{Recall}_T$) balances two components: whether an anomaly is detected at all (existence) and how completely it is covered (overlap). It is defined as:

$$\text{Recall}_T(R, P) = \frac{1}{N_r} \sum_{i=1}^{N_r} \Big[\alpha \cdot \text{ExistenceReward}(R_i, P) + (1 - \alpha) \cdot \text{OverlapReward}(R_i, P) \Big], \quad (2.22)$$

where $\alpha \in [0, 1]$ controls the trade-off between these two aspects. The ExistenceReward(R_i, P) equals 1 if any predicted interval overlaps with R_i, and 0 otherwise. The OverlapReward(R_i, P) quantifies the proportion of R_i covered by predictions, potentially incorporating positional weighting and penalizing fragmentation when multiple small predictions cover a single true interval.

Range-based precision (Precision$_T$) penalizes spurious predictions by measuring how well predicted intervals align with true anomalies:

$$\text{Precision}_T(R, P) = \frac{1}{N_p} \sum_{j=1}^{N_p} \text{OverlapReward}(P_j, R). \tag{2.23}$$

Similarly to Eq. 2.20, the series-aware F1-score (F$_{1_T}$) is calculated as the harmonic mean of Precision$_T$ and Recall$_T$.

Proximity-Aware Time Series Anomaly Evaluation

Proximity-aware time series anomaly evaluation (PATE) [136] extends evaluation beyond exact temporal alignment by introducing buffer zones around true anomalies. Pre-buffers capture early detections, while post-buffers account for delayed detections. Each predicted point is classified into one of four categories:

- True-detection: overlaps with a true anomaly interval.
- Pre-buffer detection: falls within the pre-buffer (early detection).
- Post-buffer detection: falls within the post-buffer (delayed detection).
- Outside: lies beyond all anomaly intervals and buffers (false alarm).

Undetected anomalies are categorized as either total or partial misses. PATE assigns time-varying weights $w^{\text{TP}}(t)$, $w^{\text{FP}}(t)$, and $w^{\text{FN}}(t)$ based on temporal distance from true anomalies, gradually reducing credit for detections farther from the actual event. Weighted precision and recall are computed as:

$$\text{Precision}_{e,d}(\theta) = \frac{\sum_{t=1}^{T} w^{\text{TP}}(t)}{\sum_{t=1}^{T} \left[w^{\text{TP}}(t) + w^{\text{FP}}(t) \right]}, \tag{2.24}$$

$$\text{Recall}_{e,d}(\theta) = \frac{\sum_{t=1}^{T} w^{\text{TP}}(t)}{\sum_{t=1}^{T} \left[w^{\text{TP}}(t) + w^{\text{FN}}(t) \right]}. \tag{2.25}$$

To provide threshold-independent evaluation, PATE computes weighted Precision-Recall curves across different thresholds θ for each combination of pre-buffer $e \in E$ and post-buffer $d \in D$. The area under each curve (AUC-PR$_{e,d}$) is averaged across all buffer configurations:

$$\text{PATE} = \frac{1}{|E| \cdot |D|} \sum_{e \in E} \sum_{d \in D} \text{AUC-PR}_{e,d}. \tag{2.26}$$

This approach allows for the evaluation of detection performance across multiple temporal sensitivities, thereby providing a robust assessment of timing accuracy.

2.5.2.4 *Affiliation Metrics*

Affiliation metrics [171] measure how closely predicted anomalies align with their nearest ground-truth events, providing a parameter-free, probabilistic evaluation framework.

The affiliation approach partitions the timeline based on proximity to ground-truth anomalies. Each time point is associated with its nearest ground-truth event, creating affiliation zones I_j around each true anomaly gt_j. For predictions falling within zone I_j, denoted as $\text{pred} \cap I_j$, affiliation computes directed distances that measure alignment quality:

$$D_{\text{precision},j} = \text{dist}\big((\text{pred} \cap I_j),\ gt_j\big), \tag{2.27}$$

$$D_{\text{recall},j} = \text{dist}\big(gt_j,\ (\text{pred} \cap I_j)\big), \tag{2.28}$$

where the directed distance function is defined as:

$$\text{dist}(X, Y) = \frac{1}{|X|} \int_{x \in X} \min_{y \in Y} |x - y| \, dx. \tag{2.29}$$

These distances quantify how far predicted points deviate from true anomalies (precision) and how far true anomalies are from their nearest predictions (recall).

To provide interpretable scores, affiliation converts distances into probabilities by comparing detection performance against a random baseline. The per-event precision probability is computed as:

$$P_{\text{precision},j} = \frac{1}{|\text{pred} \cap I_j|} \int_{x \in (\text{pred} \cap I_j)} F_{\text{precision},j}\big(\text{dist}(x, gt_j)\big)\, dx, \quad (2.30)$$

where $F_{\text{precision},j}$ represents the survival function obtained by sampling random points within $\text{pred} \cap I_j$. This probabilistic formulation indicates how much better the model performs compared to chance.

Overall affiliation-based precision and recall are obtained by averaging per-event probabilities:

$$P_{\text{precision}} = \frac{1}{|S|} \sum_{j \in S} P_{\text{precision},j}, \quad (2.31)$$

$$P_{\text{recall}} = \frac{1}{n} \sum_{j=1}^{n} P_{\text{recall},j}, \quad (2.32)$$

where S denotes the set of ground-truth intervals with at least one affiliated prediction, and n is the total number of ground-truth events.

2.5.2.5 *Threshold-independent Metrics*

Most anomaly detection methods produce continuous anomaly scores rather than binary labels. Converting scores to binary predictions requires selecting a threshold, introducing subjectivity and inconsistency across evaluations. Threshold-independent metrics evaluate performance across all possible thresholds, providing comprehensive assessment without arbitrary cutoff selection.

Area Under the Receiver Operating Characteristic Curve

The receiver operating characteristic (ROC) curve represents the trade-off between true positive rate (TPR) and false positive rate (FPR) across varying thresholds, reflecting the model's ability to separate anomalies from normal observations at different decision boundaries, where

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{2.33}$$

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{2.34}$$

The area under the receiver operating characteristic curve (AUC-ROC) [120] summarizes this discrimination capability in a single scalar value:

$$\text{AUC-ROC} = \int_0^1 \text{TPR}(\text{FPR})\, d\text{FPR}. \tag{2.35}$$

AUC-ROC ranges from 0 to 1, where 1 indicates perfect classification, 0.5 represents random performance, and values below 0.5 suggest systematic misclassification. Higher values indicate more effective differentiation between normal and anomalous patterns, making this metric useful for general assessment of classification capability.

Area Under the Precision-Recall Curve

In imbalanced datasets, typical of anomaly detection where normal instances greatly outnumber anomalies, precision-recall (PR) curves offer a more informative evaluation than ROC curves. By plotting Precision against Recall across thresholds, they focus on performance for the minority (anomalous) class. The area under the precision-recall curve (AUC-PR) provides a threshold-independent measure of the model's ability to achieve both high precision and recall, making it especially useful in rare-event detection.

Volume Under the Surface

While ROC and PR curves eliminate threshold dependency, they remain point-based metrics that may inadequately assess performance on continuous anomalous events. A model detecting only a single point within an extended anomaly can still achieve high area under the curve (AUC) scores, potentially misrepresenting practical effectiveness.

The volume under the surface (VUS) [41, 316] extends traditional AUC by introducing a buffer parameter that accounts for the contiguous nature of time series anomalies. Rather than evaluating performance at a single scale, VUS computes AUC across multiple buffer sizes, constructing a performance surface that captures detection quality at different temporal resolutions. This approach quantifies both temporal and amplitude differences between predicted and true anomaly signals, providing holistic assessment of detection performance.

For a buffer size b, let $\mathrm{AUC}(b)$ denote the threshold-free metric computed with that buffer (e.g., AUC-ROC(b) or AUC-PR(b)). VUS integrates these values over a range of buffer sizes:

$$\mathrm{VUS} = \int_{b_{\min}}^{b_{\max}} \mathrm{AUC}(b)\, db, \tag{2.36}$$

yielding separate formulations for ROC and PR contexts:

$$\text{VUS-ROC} = \int_{b_{\min}}^{b_{\max}} \text{AUC-ROC}(b)\, db, \tag{2.37}$$

$$\text{VUS-PR} = \int_{b_{\min}}^{b_{\max}} \text{AUC-PR}(b)\, db. \tag{2.38}$$

In practice, buffer sizes are sampled discretely, and the integrals are approximated through summation:

$$\text{VUS-ROC} \approx \sum_{i=1}^{N_b} \text{AUC-ROC}(b_i)\, \Delta b_i, \tag{2.39}$$

$$\text{VUS-PR} \approx \sum_{i=1}^{N_b} \text{AUC-PR}(b_i)\, \Delta b_i, \tag{2.40}$$

where $\{b_i\}_{i=1}^{N_b}$ represents the selected buffer sizes and Δb_i denotes the spacing between consecutive values. Higher VUS values indicate better performance, as they reflect consistently high detection quality across multiple temporal resolutions and buffer configurations, providing a robust assessment of how well predictions align with true anomaly signals.

2.5.2.6 *Operational Metrics*

While traditional metrics evaluate overall detection accuracy, operational metrics assess practical deployment considerations such as event detection rate, detection speed, false alarm frequency, and ranking quality. These metrics are critical for streaming anomaly detection in real-time systems where timely, accurate responses are essential.

Event Detection Rate

The event detection rate (EDR) measures the proportion of distinct anomaly events that are successfully detected. An anomaly event is defined as a contiguous sequence of anomalous points in the ground-truth signal y_{true}. An event is considered detected if at least one point within its duration is flagged as anomalous in the predicted sequence y_{pred}. Formally, let $\mathcal{E} = \{E_1, E_2, \ldots, E_m\}$ denote the set of m ground-truth anomaly events. Then:

$$\text{EDR} = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\left(\exists\, t \in E_i : y_{\text{pred}}(t) = 1\right), \tag{2.41}$$

where $\mathbb{1}(\cdot)$ is the indicator function returning 1 if the i-th event is detected and 0 otherwise. The EDR thus quantifies event-level recall rather than point-level accuracy, emphasizing whether each anomalous episode is detected at least once. Higher EDR values indicate stronger coverage of true anomaly events, even if only partially detected.

Mean Time to Detect

The mean time to detect (MTTD) quantifies the average delay between anomaly occurrence and detection, measuring system responsiveness. This metric is essential when detection speed directly impacts operational outcomes.

For n detected anomalies, MTTD is computed as:

$$\text{MTTD} = \frac{1}{n}\sum_{i=1}^{n}(T_{\text{detect},i} - T_{\text{true},i}), \tag{2.42}$$

where $T_{\text{true},i}$ denotes when the i-th anomaly occurred and $T_{\text{detect},i}$ denotes when it was first detected. Lower MTTD values indicate rapid detection, enabling prompt response to important events.

To provide sampling-frequency-agnostic evaluation, MTTD can alternatively be expressed in observation units rather than temporal units:

$$\text{MTTD}_{\text{obs}} = \frac{1}{n}\sum_{j=1}^{n}(\text{idx}_{\text{detect},j} - \text{idx}_{\text{true},j}), \tag{2.43}$$

where $\text{idx}_{\text{true},j}$ represents the sequence index where the j-th anomaly begins, and $\text{idx}_{\text{detect},j}$ is the index of the first detected anomalous point after that onset.

False Alarm Rate

In continuous monitoring systems, excessive false alarms diminish operational utility and lead to alert fatigue. The false alarm rate (FAR) quantifies the frequency of spurious alerts over time:

$$\text{FAR} = \frac{\text{\# false positives in time window}}{\text{duration of time window}}. \tag{2.44}$$

The FAR is typically expressed as false alarms per hour or per day. Lower FAR values indicate fewer spurious alerts, improving system trustworthiness and reducing unnecessary investigation costs.

Precision-at-k

Precision-at-k (Precision@k) evaluates the quality of top-ranked detections. Given predictions ordered by anomaly score, this metric measures the proportion of true anomalies among the top k flagged instances:

$$\text{Precision@k} = \frac{|\text{true anomalies} \cap \text{top-}k \text{ predictions}|}{k}. \tag{2.45}$$

Higher Precision@k values indicate effective concentration of true anomalies at the top of the ranking, enabling analysts to prioritize investigations efficiently and minimize wasted effort on false alarms.

3 RELATED WORK

While centered on generative models and temporal representations, this research contributes broadly to time series data mining and artificial intelligence for IT operations (AIOps) in financial infrastructures. This chapter provides an overview of foundational work across four interconnected areas: time encoding methods, deep learning architectures for forecasting, deep learning approaches to anomaly detection, and AIOps techniques for failure management [50, 75, 234, 235, 311, 466].

Section 3.1 examines methods for capturing and representing temporal information, progressing from handcrafted features to learnable embeddings. Section 3.2 surveys deep learning architectures for time series forecasting, presenting the main architectural paradigms. Section 3.3 reviews deep learning approaches to anomaly detection, categorizing methods by their underlying principles. Finally, Section 3.4 explores the emerging field of AIOps, focusing on failure management techniques and explainable anomaly detection methods.

3.1 TIME ENCODING FOR TIME SERIES

In many real-world domains, such as finance [228], retail [189, 340, 371], and weather forecasting [300, 307], temporal data accompany observations. Incorporating temporal information into models helps capture long-range dependencies and cyclical patterns, enabling more accurate predictions [192].

Time encoding methods capture and represent temporal information by treating time embeddings as independent features. Early approaches relied on handcrafted features tailored to the downstream tasks [31, 73, 212]. These methods typically involved decomposing timestamps into components (e.g., month, day, hour), assigning an embedding to each component, and summing these embeddings to

obtain the final time representation [418, 431], or appending temporal attributes, such as sine and cosine encodings for cyclical variables [58].

While simple, handcrafted methods are resource-intensive, require domain-specific expertise, and are limited to capturing only predefined temporal patterns [192]. Consequently, they often fail to capture the full complexity of temporal relationships, particularly in data exhibiting seasonal variations or irregular patterns [65]. Subsequent research, progressing from handcrafted features to learnable representations, has exploited and extended the concept of positional encoding introduced in Transformer architectures [412]. These techniques, primarily applied in natural language processing [290, 320, 416], have been adapted for continuous time representation in time series [268, 442].

Functional time encoding (FTE) are a generalized version of time encoding methods that offer advantages such as better management of long-range dependencies and adaptive weighting of temporal information. They transform temporal input into multi-dimensional embeddings using predefined linear and nonlinear transformations. Representative works include Time2Vec [192] and Functional Time Representation [441]. Time2Vec extends the concept of positional encoding for continuous time representation in time series through neural decompositions [133, 138], using learnable frequencies and phases to adaptively capture multiple periodicities. In contrast, Xu et al. [441] introduce a functional feature map that incorporates the time interval between events into high-dimensional spaces.

Time encodings provide a general-purpose representation of time applicable to existing models such as recurrent neural networks (RNNs) and long short-term memorys (LSTMs) [143] or integrated into attention-based architectures [431]. However, when time series exhibit structural irregularities, FTE methods show limited expressive capacity effectively to model such patterns due to their reliance on predefined transformations primarily designed for periodic components. To overcome these limitations, emerging research directions draw inspiration from deep function learning [271], employing learnable nonlinear transformations to adaptively model temporal information and capture irregular trends, sudden changes, and overlapping periodicities [65].

3.2 TIME SERIES FORECASTING WITH DEEP LEARNING

Temporal modeling is fundamental to practical applications, spanning domains from climate [294] and healthcare [404] to retail optimization [44] and financial markets [9]. Classical approaches rely on parametric models grounded in domain expertise, including AR processes [46], exponential smoothing techniques [132, 426], and structural time-series models [153]. In contrast, contemporary machine learning methods offer data-driven alternatives that automatically discover temporal patterns [5], becoming increasingly vital as data availability and computational resources expand.

Deep learning has emerged as a promising approach, motivated by major achievements in reinforcement learning [370], computer vision [210], and natural language processing (NLP) [97]. Neural networks incorporate architectural inductive biases [30] that reflect dataset characteristics, enabling automatic learning of complex representations [34] without extensive manual feature engineering. The proliferation of open-source frameworks [2, 318] has further democratized deep learning by simplifying training procedures and enabling flexible customization of network architectures and optimization objectives.

Most deep learning approaches frame forecasting as supervised learning over sliding windows [213, 313]. Deep neural networks learn predictive relationships through nonlinear transformations that construct intermediate representations [34]. An encoder $g_{\text{enc}}(\cdot)$ compresses historical information into a latent representation $\mathbf{z}_t$, which a decoder $g_{\text{dec}}(\cdot)$ then uses to generate predictions. The choice of architecture determines what temporal patterns the model can capture.

Multilayer Perceptrons

The concept of artificial neural networks (ANNs), first introduced by McCulloch and Pitts [289], was inspired by biological neural processing, with neurons processing information in parallel. MLP represents the most fundamental feed-forward architecture, consisting of three components: an input layer, one or more hidden layers, and an output

layer. Network depth, determined by the number of hidden layers, distinguishes shallow from deep learning models.

Each layer contains neurons with trainable connection weights that the learning algorithm updates to map input-output relationships. Unlike recurrent architectures, MLPs maintain only forward connections between neurons. Early applications to time-series forecasting in the 1990s [162] demonstrated their potential as universal function approximators, offering the flexibility to adapt to data without predefined assumptions. MLPs handle multivariate inputs and outputs flexibly but are constrained by static mappings [389]. However, these studies also highlighted two critical challenges: determining optimal network structure and hyperparameters remains complex with substantial performance impact, and rigorous validation procedures are essential to establish advantages over classical methods.

Preprocessing steps significantly influence performance, with simpler models using carefully selected input variables often outperforming complex architectures. Ensemble approaches combining multiple MLPs have demonstrated improved accuracy. A fundamental limitation is that MLPs treat each input independently, failing to capture temporal ordering. This motivates specialized architectures like RNNs and convolutional neural networks (CNNs) that explicitly encode temporal structure [351].

However, recent work has challenged the necessity of complex sequential architectures, with MLP-based models like TSMixer [62] and TiDE [91] demonstrating that simple designs with effective mixing operations can match or outperform state-of-the-art recurrent and attention-based models on several benchmarks.

Convolutional Neural Networks

Originally designed for image data [210], CNNs [219] have been adapted for time-series through causal convolutions [43, 312], defined as convolutional filters that use only past information for predictions. CNNs excel at extracting features from high-dimensional grid-structured data through the convolution operation: a sliding filter that creates feature maps capturing repeated patterns across

different data regions. This distortion invariance, i.e., the ability to extract features regardless of position, makes CNNs suitable for one-dimensional sequential data [43].

The architecture typically comprises convolutional layers for feature extraction, pooling layers for spatial reduction, and fully-connected layers for aggregation. Three key principles distinguish CNNs: local connectivity (each neuron connects only to a local receptive field), shared weights (neurons in the same layer share convolution parameters), and translation equivariance. These properties substantially reduce trainable parameters compared to fully-connected networks, improving training efficiency [43]. Stacking multiple convolutional layers enables the network to learn multi-scale temporal representations [450].

A key limitation is that CNNs can only access information within their receptive field (i.e., the lookback window determined by the network architecture). To efficiently capture long-range dependencies without excessive parameters, modern architectures employ dilated convolutions [22], which exponentially increase the temporal span covered by each successive layer. The WaveNet architecture [312] exemplifies this approach, enabling coverage of 2^l time steps at layer l.

Inspired by WaveNet [312], temporal convolutional networks (TCNs) [22] represent a specialized CNN architecture for sequential modeling. TCNs employ two distinguishing features: causal convolutions prevent information leakage from future time steps, and the architecture processes sequences of arbitrary length while producing equal-length outputs. Dilated causal convolutions expand the receptive field without resolution loss (eliminating pooling requirements) [464], while residual connections enable increased depth for handling extensive historical contexts.

Comparative studies [22] demonstrate several TCN advantages over generic RNNs: lower memory requirements through shared convolutional filters, parallel processing of long input sequences (versus sequential RNN computation), and more stable training that avoids vanishing gradients. These characteristics have driven increasing TCN adoption for time-series forecasting applications.

Recurrent Neural Networks

RNNs [106] were developed as a temporal variant of ANNs that connect each time step with previous observations to model temporal dependencies, providing native support for sequential data with dynamic dependency learning [284, 331, 423]. The network processes one observation at a time, learning both patterns between inputs and outputs and internal patterns within the observation sequence. Unlike MLPs that ignore temporal relationships, RNNs maintain memory of prior information and its relevance for forecasting.

Unlike CNNs, RNNs do not require an explicit lookback window; they theoretically have infinite memory. At each time step, the network updates its hidden state $\mathbf{z}_t$ based on the previous state $\mathbf{z}_{t-1}$ and current inputs. Early implementations in the late 1980s explored various approaches to providing neural networks with memory [100]. The Jordan RNN [185] inspired the Elman RNN [106], which forms the foundation of modern recurrent architectures.

RNNs have been extensively applied to sequence modeling [140], achieving strong results in NLP [461] and success across diverse forecasting applications including financial markets [199], wind speed prediction [463], and solar radiation forecasting [421], with notable performance in competitions such as M4 [281].

Early RNN variants suffered from vanishing and exploding gradients [35, 140, 164], limiting their ability to learn long-range dependencies. LSTM networks [165] address these issues through a gating mechanism that controls information flow. LSTMs use three gates (input, output, and forget gates) to regulate what information to store, use, or discard from a separate cell state. This architecture enables stable learning over extended temporal sequences and has become the standard recurrent unit for time-series forecasting [331, 349]. Variants such as echo state networks (ESNs) [179] and gated recurrent unit (GRU) [72] offer alternative gating mechanisms with different computational trade-offs.

RNNs share conceptual similarities with Bayesian filters [350]: both recursively update a hidden state over time, with RNNs learning to approximate state transition and error correction steps simultaneously through data-driven training [255].

Attention Mechanisms and Transformers

Attention mechanisms [20, 72] advanced sequence modeling by enabling networks to dynamically focus on relevant time steps, regardless of temporal distance. Unlike RNNs that compress all history into a fixed-size state, attention computes weighted combinations of past features, where weights are generated based on the relevance of each time step to the current prediction.

Transformer architectures [97, 412] leverage self-attention to achieve state-of-the-art performance across multiple domains and have been successfully adapted for time-series forecasting [238, 254, 326, 363]. In forecasting applications, attention-based models have demonstrated superior performance over comparable recurrent networks. Attention provides two key advantages: models can directly access significant past events (e.g., holidays or promotions in retail forecasting), and they can learn regime-specific temporal dynamics by applying distinct attention patterns to different contexts [254].

A key limitation of standard attention mechanisms is their quadratic computational complexity with respect to sequence length, which becomes prohibitive for extended temporal sequences. To address this scalability challenge, several efficient variants have been proposed: LogTrans employs LogSparse attention [238], Reformer uses locality-sensitive hashing [204], Informer introduces ProbSparse attention to focus on dominant queries [486], and Autoformer replaces attention with autocorrelation mechanisms [432].

Mixture of Experts

MoE architectures decompose learning tasks across specialized expert networks, each focusing on different aspects of the data. Originally introduced by Jacobs et al. [178], MoE models employ a gating mechanism to dynamically route inputs to appropriate experts, whose outputs are subsequently aggregated. This modular structure enables efficient scaling and specialization, particularly valuable for capturing diverse temporal patterns within complex datasets.

The gating network determines expert selection through various strategies. Dense MoE variants compute weighted combinations of all expert outputs, while sparse approaches activate only a subset. Shazeer et al. [359] proposed Sparsely-Gated MoE with thousands of feed-forward subnetworks, achieving substantial capacity increases with minimal computational overhead. Subsequent work improved stability and efficiency [223, 227], with V-MoE [223] incorporating sparse activation within vision transformer architectures.

For multi-task scenarios, Ma et al. [276] demonstrated that shared expert weights across tasks improve generalization, while task-specific experts capture domain-particular patterns. This principle extends naturally to time series, where experts can specialize in different temporal regimes, seasonal behaviors, or external conditions.

Sleeping experts represent a specialized variant where only certain experts produce outputs under predetermined conditions [40, 127]. Devaine et al. [96] applied this concept to electricity demand forecasting, using sequential convex aggregation to weight active experts based on calendar conditions such as working days, holidays, or seasonal variations. This deterministic gating based on temporal conditions differs from learned gating mechanisms but offers interpretability through explicit expert-condition associations.

MoE architectures provide several advantages for time series modeling: experts specialize in distinct temporal patterns or seasonal behaviors, the gating mechanism offers interpretability by revealing pattern-expert associations, and sparse activation reduces computational overhead while maintaining model capacity for complex dependencies. Recent applications demonstrate improved forecasting accuracy with reduced parameter counts [266, 362]. In financial applications, MoE models support anomaly detection and pattern recognition for fraud monitoring [186].

Graph Neural Networks

Graph neural networks (GNNs) enable spatiotemporal forecasting by explicitly modeling relationships between multiple time series as graph structures [70, 244, 436, 462]. While deep learning models

like LSTNet [213] and TPA-LSTM [363] capture nonlinear temporal patterns effectively, they often overlook or model only implicitly the rich dynamic spatial correlations between time series. GNN-based approaches address this limitation by explicitly incorporating graph structures that delineate the strength of interconnections between variables [184].

GNN-based forecasting models operate by learning both spatial dependencies (inter-variable relationships) and temporal dependencies (intra-variable patterns). Spatial modeling typically employs spectral GNNs, spatial GNNs, or hybrid approaches to capture correlations between time series. Temporal modeling captures dependencies within individual time series through recurrent, convolutional, or attention-based mechanisms in either time or frequency domains [183].

The fusion of spatial and temporal modules can follow different architectural patterns: discrete factorized models process spatial and temporal dependencies independently, discrete coupled models integrate both within a unified process, while continuous models employ neural differential equations to abstract the underlying dynamics [74, 119].

Composite and Hybrid Models

Despite deep learning's success in many domains, pure neural approaches historically underperformed simple statistical methods in forecasting competitions [175, 282]. Two key issues were identified: overfitting in low-data regimes [280], and sensitivity to input preprocessing [349, 357].

TBATS [93] automates the modeling of complex, evolving seasonality by integrating Fourier representations with exponential smoothing and Box-Cox transformations. While potentially computationally intensive, this state-space approach offers greater flexibility than rigid harmonic regression by allowing seasonal patterns to change over time. Prophet [397] further democratizes forecasting through configurable, interpretable frameworks accessible to domain experts.

Hybrid statistical-neural models [331, 375] address these limitations by combining classical time-series models with deep learning, typi-

cally by using neural networks to generate parameters for statistical models at each time step. This approach incorporates domain knowledge through well-established statistical structures while leveraging deep learning's capacity to learn complex patterns. The ES-RNN [375], winner of the M4 competition, exemplifies this strategy by using exponential smoothing [426] to capture non-stationary trends while an RNN learns additional effects.

Hybrid models can be non-probabilistic (producing point forecasts by combining statistical equations with neural network outputs) or probabilistic (using neural networks to generate parameters for predictive distributions). For instance, deep state space models [331] employ neural networks to produce time-varying parameters for linear state space models, performing inference via Kalman filtering [187]. This separation of stationary and non-stationary components reduces preprocessing requirements and improves generalization, particularly for small datasets.

Beyond statistical-neural hybrids, architectural hybrids combine multiple deep learning components to leverage complementary strengths [195, 338]. LSTNet [213] integrates CNNs for local feature extraction with recurrent layers for long-term dependencies, while DeepAR [349] combines autoregressive structures with RNNs for probabilistic forecasting.

Generative Models

Generative modeling offers an alternative paradigm for time-series forecasting by learning to synthesize realistic temporal patterns. GANs [141] have been adapted to sequential data by integrating temporal architectures within adversarial training frameworks, where a generator produces synthetic sequences while a discriminator distinguishes real from generated data.

TimeGAN [458] integrates unsupervised adversarial learning with supervised autoregressive modeling to maintain temporal coherence in generated sequences. The architecture processes static features (time-invariant characteristics) and temporal features through an embedding network producing latent codes. Supervised loss en-

forces consistency between real and synthetic latent representations, while adversarial training ensures distributional fidelity. Evaluations across financial markets, energy consumption, and medical event data demonstrated improvements over earlier temporal GANs variants such as RCGAN [110] and C-RNN-GAN [292].

Long sequences present computational challenges due to dimensionality growth. SigCWGAN [252] mitigates this through the Signature Wasserstein-1 metric, capturing temporal dependencies with reduced computational cost compared to standard Wasserstein distances. Its conditional autoregressive feed-forward generator maps historical observations and noise to future sequences, achieving strong results on financial indices.

Domain-specific objectives can be incorporated through decision-aware frameworks. DAT-CGAN [387] employs multi-Wasserstein losses over decision-relevant quantities for financial portfolio optimization, jointly training discriminators on both raw returns and portfolio statistics. Additional approaches include QuantGANs [425] for financial approximation and hybrid architectures combining LSTM, CNN, and MLP components [169, 237].

Despite advances in capturing distributional properties and long-range dependencies, GAN-based forecasting faces notable limitations. Training stability remains challenging compared to discriminative approaches, particularly for extended sequences [50]. Moreover, adoption in forecasting applications lags behind computer vision domains [422]. Current research increasingly explores GANs for data augmentation and scenario generation rather than direct point forecasting, where they complement discriminative models.

3.3 TIME SERIES ANOMALY DETECTION WITH DEEP LEARNING

Time series anomaly detection methods are generally categorized into three main groups: statistical approaches, classical machine learning techniques, and deep learning methods [13]. Statistical approaches establish probabilistic models, such as AR and Moving Average models [37], offering simplicity and theoretical grounding but relying

on restrictive assumptions about data distributions and linear correlations that may not hold in complex scenarios. Classical machine learning methods, including Local Outlier Factor (LOF) [48], DBSCAN [111], and One-Class support vector machine (SVM) [355], learn patterns from subsets of time series data to make predictions on unseen portions. However, both traditional statistical and classical machine learning approaches struggle to capture the complex structural relationships and temporal dependencies inherent in large-scale, high-dimensional time series data [471]. Deep learning has emerged as a powerful alternative, demonstrating superior performance across diverse scenarios [15, 466]. Deep anomaly detection methods exhibit robust linear and nonlinear modeling capabilities, effectively handling data scalability challenges while extracting temporal features from complex datasets [267, 342, 480].

Deep learning methods for time series anomaly detection exploit different principles to distinguish normal from anomalous patterns. Many approaches repurpose architectures originally developed for time series forecasting (see Section 3.2) by analyzing discrepancies between predicted and actual values. These forecasting-based methods provide intuitive interpretability and excel in real-time scenarios where normal patterns are abundant [172], though they struggle with rapid changes and extended anomalous sequences [445].

Reconstruction-based methods learn compressed representations through encoder-decoder architectures, producing high errors for unseen anomalies. They generalize well across anomaly types and provide interpretable error signals for fault localization [446], but risk overfitting when trained on contaminated data, potentially learning to reconstruct the very anomalies they should detect [14].

A pervasive challenge across all paradigms is class imbalance, where anomalies constitute only a small minority of observations. Generative methods address this by synthesizing samples to augment limited training data [125, 256]. Beyond augmentation, generative approaches are increasingly integrated directly into detection pipelines. VAEs provide likelihood-based scoring with statistical grounding, while GANs capture high-order distributional structures [141, 201], though both require careful hyperparameter tuning and can face training instabilities [353].

Density-based approaches employ normalizing flows (NFs) to model complex distributions through invertible transformations, enabling rigorous probabilistic anomaly assessment [88]. However, they face the curse of dimensionality and demand substantial clean training data to avoid distorted probability estimates [305]. Contrastive learning methods discover discriminative features through self-supervised comparison of augmented data views, proving particularly effective when labeled anomalies are scarce [148]. Their performance, however, depends critically on augmentation strategy design [452].

Forecasting-based Approaches

Deep learning architectures originally developed for time series forecasting can be effectively adapted for anomaly detection. These methods operate on the principle that prediction errors remain small for normal patterns but increase significantly for anomalous sequences. The anomaly detection process involves computing an anomaly score based on the discrepancy between predicted and actual values, which is then compared against a predetermined threshold. Threshold selection presents a critical challenge, as different thresholding strategies significantly impact detection performance [172, 233, 382].

Forecasting architectures leverage their inherent temporal modeling capabilities for anomaly detection. Recurrent architectures, such as LSTMs and GRUs, capture long-term dependencies through internal states, enabling detection of anomalies that deviate from extended temporal patterns [255]. Ergen et al. [108] combined LSTM with one-class SVM for variable-length multivariate sequences, while AQADF [261] employs hybrid attention mechanisms to capture both global trends and local fluctuations. Despite their effectiveness, RNNs face computational limitations due to sequential processing and potential gradient instability with very long sequences [343].

CNN-based methods, particularly TCNs with dilated convolutions, offer computational efficiency through parallel processing while maintaining large receptive fields [312]. DeepAnT [297] demonstrates this approach by combining a TCN-based predictor with an anomaly

detector for large-scale unlabeled data, while CAD [366] enhances multivariate detection through multi-task learning. Empirical studies show that well-tuned CNNs can match or outperform more complex architectures on several benchmarks [304], though they require predetermined window sizes and assume temporal stationarity [118, 379].

GNN-based methods extend forecasting to multivariate scenarios by explicitly modeling spatial inter-variable dependencies alongside temporal dynamics. GDN [95] pioneered attention-based graph structures for anomaly detection, while subsequent methods incorporated causal modeling: CGAD [121] employs transfer entropy for causal graph construction, and GCAD [269] leverages Granger causality to identify deviations in causal relationships. Hybrid architectures such as HGTMA [130] integrate graph convolutional networks (GCNs) with Transformers for comprehensive spatiotemporal modeling, while Graph-MoE [170] uses mixture-of-experts frameworks for adaptive pattern capture. Although these methods provide enhanced interpretability through attention mechanisms and causal analysis, they introduce significant computational complexity [364] and require meaningful inter-variable relationships to perform effectively [407].

Reconstruction-based Approaches

Reconstruction-based methods learn the distribution of normal time series data by encoding inputs into a compressed latent representation and subsequently reconstructing them. These models are typically trained exclusively on normal data, resulting in low reconstruction errors for normal patterns but high errors for anomalous inputs. The reconstruction error, commonly measured using MSE, serves as the anomaly score. These approaches rely primarily on AEs, GNNs, and Transformer architectures.

AEs comprise an encoder that compresses time series segments into lower-dimensional latent representations and a decoder that reconstructs the original input. Various architectures including CNNs,

RNNs, and fully connected networks (i.e., MLP) have been employed to implement AEs for anomaly detection.

Early approaches established foundational techniques using standard architectures. MSCRED [472] constructs multi-scale feature matrices through convolutional encoders with ConvLSTM and attention mechanisms to model inter-sensor correlations. Kieu et al. [198] propose ensembles of sparsely-connected recurrent AEs for temporal feature extraction. Subsequent methods enhanced robustness through adversarial training strategies, including USAD [14] and APAE [142]. Architectural refinements have focused on improved temporal modeling: RAMED [360] and TSAE [302] adapt recurrent structures by varying decoding lengths and separating short- and long-term components. CAE-Ensemble [55] and PAMFE [473] leverage probabilistic AE architectures to capture both global trends and local details. SVD-AE [456] integrates singular value decomposition with asymmetric AE architectures for modeling complex multivariate patterns.

Advances include dual-branch designs and optimization-based approaches. Song et al. [378] and AEVAE [154] employ genetic algorithm-based optimization and dual-branch architectures to refine reconstruction quality. These AE-based methods offer simplicity and interpretability through direct reconstruction error measurements [25]. However, they face challenges, such as overfitting when training data are insufficiently curated, potentially learning to reconstruct anomalies [105], and a failure to capture long-range dependencies critical for global context modeling [239].

Recent developments integrate GNNs and Transformer architectures to capture inter-variable relationships and long-range temporal dependencies. STGAT-MAD [469] combines GNN and LSTM components for joint spatial-temporal modeling. MIXAD [200] introduces memory networks that preserve normal spatiotemporal patterns through memory attention mechanisms, enhancing both detection performance and interpretability.

Transformer-based approaches have demonstrated significant advances in modeling global context. Anomaly Transformer [446] employs self-attention mechanisms with specialized Anomaly-Attention modules to distinguish normal and anomalous patterns. PAFormer [21] utilizes parallel attention through Global Enhanced

Representation and Local Perception modules to learn global-local distribution variance. MAN-QSM [483] leverages pairwise associations and sequence-level information via anomaly masking mechanisms. Uni-AD [157] and TranAD [408] combine self-attentive autoregression with adversarial and meta-learning strategies for improved robustness across variable-length inputs.

Hybrid architectures offer enhanced accuracy and robustness by effectively capturing both local details and long-range dependencies [130]. GCFormer [438] discovers causal spatiotemporal relationships through attention mechanisms, enabling rapid identification of critical series during anomalous events. MEMTO [377], EdgeConvFormer [263], and ADFormer [468] incorporate memory-guided updates, dynamic graph convolution, and fusion learning to handle overgeneralization and subtle pattern variations. GDFormer [265] proposes global dictionary representations to address limited receptive field limitations in standard Transformers.

While traditional approaches provide simplicity and direct anomaly scoring, they may lack global context modeling capabilities [25]. These architectures excel at modeling global context and complex inter-variable dependencies, with Transformers providing enhanced long-range modeling and graph modules offering improved interpretability through explicit spatial dependency representation. However, the performance comes at increased computational cost, requiring careful hyperparameter tuning and substantial computational resources during training and inference.

Generative-based Approaches

Generative models learn the underlying distribution of normal time series data to identify anomalies through probabilistic modeling. Unlike deterministic reconstruction-based methods that focus on the magnitude of reconstruction residuals, generative approaches explicitly model the data generation process, computing anomaly scores based on likelihood estimates or reconstruction probabilities [201]. The two primary architectures are VAEs and GANs.

VAE-based methods employ probabilistic encoder-decoder architectures trained with the ELBO, balancing reconstruction accuracy with latent space regularization through KL-divergence. Representative approaches include OmniAnomaly [386], which combines stochastic recurrent networks with GRU cells and planar normalizing flows, and GRELEN [479], which integrates GNNs for inter-sensor relationships. GGM-VAE [147] employs Gaussian Mixture priors to better characterize multimodal distributions. Domain-specific methods demonstrate effectiveness in applications ranging from IoT security [173] to biomedical signal analysis [459]. VAEs provide robust probabilistic scores, prevent overfitting through latent regularization [98], and support adaptive thresholding [317]. However, balancing reconstruction and KL-divergence terms requires careful tuning, and overly smooth reconstructions may reduce sensitivity to minor anomalies [47].

GAN-based methods employ adversarial training where generators produce synthetic normal samples while discriminators distinguish real from generated data. Predominantly inspired by AnoGAN [353], time series methods mainly use LSTM-based architectures. GAN-AD [231] models multivariate sensor data while employing principal component analysis (PCA) for dimensionality reduction, MAD-GAN [232] combines discrimination and reconstruction losses across multiple window sizes, and TadGAN [134] introduces cycle consistency loss for improved reconstruction. Alternative architectures include BeatGAN [485], which employs one-dimensional CNNs for enhanced robustness in ECG analysis, DCT-GAN [243], which combines dilated CNNs with Transformers for multi-scale temporal capture, and transformation-based approaches like PowerPlantGAN [77], which converts time series to distance images. Additional methods incorporate ensemble techniques. LSTM-GAN-XGBoost [448] combines adversarial training with gradient boosting for real-time detection, while MARU-GAN [287] employs seq2seq architectures to capture extended historical context for collective anomalies. GANs yield sharper reconstructions than VAEs [141] and detect subtle anomalies through discriminative learning [134]. However, they face training instability with potential mode collapse [347], sensitivity to sliding window parameters [26], and increased computational costs.

Hybrid architectures leverage complementary strengths from multiple paradigms. LSTM-VAE-GAN [309] integrates VAE encoders with adversarial training to generate low-dimensional embeddings for improved reconstruction. TAnoGAN [29] combines LSTM-based generation with discriminative latent features, detecting anomalies through reconstruction loss and distribution conformity. DAEMON [66] enhances robustness by constraining autoencoder outputs with adversarial components, while DiffGAN [430] incorporates learnable denoising processes beyond fixed diffusion schedules [163]. Methods integrating GNN components [76, 239] further capture inter-variable relationships alongside temporal dynamics. The choice between these approaches requires balancing training stability, computational cost, and temporal characteristics of the target application.

Density-based Approaches

Density-based anomaly detection methods model the probability distribution of time series data and identify anomalies as observations occurring in low-density regions. The fundamental principle is to estimate the likelihood of observations and flag those with abnormally low probability as anomalous, leveraging the distinct statistical characteristics that separate anomalies from normal patterns.

NFs have emerged as a significant framework for density estimation in this context. NF constructs an invertible transformation that maps complex data distributions to simpler base distributions, typically multivariate Gaussians. This enables direct probability assessment of observations. During training, the model learns to maximize the likelihood of normal data; during inference, observations with likelihood below a threshold are classified as anomalies.

Probabilistic modeling has been integrated with deep architectures to enhance detection capabilities. NSIBF [123] combines LSTM networks with Bayesian filtering to track latent state uncertainties over time. For multivariate time series, several methods leverage graph structures to capture inter-variable dependencies. GANF [88] employs Bayesian networks to model static dependency structures, whereas MTGFlow [487] introduces dynamic graph structure learn-

ing to adapt to evolving relationships and entity-specific characteristics. GNF [277] integrates Transformers with GNNs within the NF framework to capture both temporal dynamics and feature correlations. AFNF [144] applies decomposition strategies and attention mechanisms to model conditional densities over time series components explicitly. Additionally, SCNF [451] combines graph neural networks with GRUs to jointly optimize imputation and detection, improving robustness.

Density-based approaches offer rigorous probabilistic anomaly scoring with strong theoretical foundations [48], naturally handling complex multimodal distributions and subtle anomalies that simpler methods may overlook. The integration of advanced architectures such as Transformers and GNNs effectively addresses spatiotemporal complexities in multivariate settings [76]. However, these methods face significant computational demands, particularly with NF models requiring complex architectures and iterative transformations [203]. They also require substantial amounts of clean training data, as contamination can distort learned distributions and degrade performance [305]. Furthermore, despite their statistical rigor, density-based methods may provide less intuitive explanations compared to reconstruction-based approaches, potentially limiting interpretability for domain experts [249].

Contrastive Learning-based Approaches

Contrastive learning approaches [148] apply self-supervised learning principles to anomaly detection by learning discriminative representations through comparison of data transformations. The fundamental strategy involves generating multiple views of the same time series through data augmentation: positive pairs consist of the original series and its lightly modified versions, while negative pairs include heavily distorted or shuffled segments. The learning objective encourages the model to produce similar embeddings for positive pairs while maximizing distance between negative pairs. This process creates a representation space where normal patterns cluster together, while anomalies are positioned farther apart. The contrastive loss it-

self can serve as an anomaly score, with normal observations yielding low loss values and anomalies producing higher scores.

Several architectures have been developed within this framework. DCDetector [452] employs a dual-attention mechanism in a Transformer architecture to enhance separation between normal and anomalous patterns without requiring reconstruction objectives. ContrastAD [229] leverages anomaly-induced temporal transformations to enforce distinct latent representations for different pattern types. TriAD [388] extends this approach by incorporating self-supervised learning across time, frequency, and residual domains to capture normal data characteristics more comprehensively. TiCTok [188] introduces a specialized token encoder that transforms raw time series into latent embeddings, using contrastive loss directly for anomaly scoring. PCRTA [392] improves detection by training models to identify various temporal disturbances, operating on the principle that samples difficult to classify are more likely anomalous.

The primary advantage of contrastive learning methods lies in learning from unlabeled data, extracting robust features that capture subtle distinctions between normal and anomalous patterns [64, 156]. The contrastive loss provides a natural anomaly score, simplifying the detection pipeline. However, performance depends heavily on augmentation design and negative sample selection [196, 435]. Poorly designed augmentations can fail to capture relevant pattern variations, while inadequate negative sampling may not provide sufficient discriminative signal. Additionally, processing numerous positive and negative pairs demands substantial computation and memory, particularly for long sequences or high-dimensional data. Despite these challenges, contrastive approaches demonstrate strong potential for anomaly detection when labeled data are limited [167], though designing augmentation strategies and negative sampling remain active research areas [90].

3.4 ARTIFICIAL INTELLIGENCE FOR IT OPERATIONS

IT system monitoring inspects measurable events and outputs to determine proper operation. Traditional monitoring relied on expert-

defined thresholds, where exceeding these values indicated unexpected behavior. However, the scale and complexity of modern IT operations have rendered threshold-based methods impractical, necessitating automated monitoring.

Real-world systems generate large volumes of log data continuously, encompassing functional logs (application functionality) and non-functional logs (system operations, performance, security). These data are inherently dynamic, generated during process execution and dependent on runtime values, with new logs frequently added during development cycles. Log data are typically unlabeled, presenting challenges for traditional monitoring approaches.

Artificial intelligence for IT operations (AIOps) lacks a universally accepted definition, remaining a largely unstructured, cross-disciplinary area. Despite definitional variations, AIOps commonly aims to provide complete visibility into IT operational states, thereby enhancing customer services [311]. The field emerged from the intersection of machine learning, big data, streaming analytics, and IT operations management [224], driven by the need to prevent and resolve issues proactively while reducing manual effort in tasks such as monitoring, anomaly detection, RCA, and recovery. These approaches infer a system's internal state from external outputs, a concept known as observability [89, 306]. Dedicated roles such as site reliability engineers [36] and DevOps practitioners [221] have emerged to cope with increasing complexity, though automated approaches still face limitations in adaptability and scalability when handling large volumes of real-time data. Current AIOps solutions depend on training data quality and require trust in tools whose outputs may lack explainability, while introducing infrastructure complexity and incurring high implementation and maintenance costs [311].

Research in this field typically focuses on failure management [205, 295], which minimizes failure occurrence and impact, and resource provisioning [89, 306], which optimizes allocation. This work focuses on failure management through proactive (failure avoidance [236, 437]) and reactive (failure tolerance [311]) approaches. Failure tolerance encompasses detection [14, 386, 445], RCA [246, 258], and remediation [257, 420]. Anomaly detection is the predominant approach for failure identification [310], assuming failures manifest as

irregular patterns across observable indicators such as key performance indicators (KPIs), metrics, logs, and traces. However, when an anomaly detection system triggers an alert, the operator requires an explanation.

A comprehensive explanation should address anomaly localization, diagnosis, and remediation. Techniques addressing at least one aspect are defined as explainable anomaly detection (XAD) techniques [249], overlapping with XAI [99]. While explanations are straightforward with traditional statistical methods, they become considerably more complex with nonlinear models, where even developers may struggle to trace decision paths. Systems that provide understanding to designers are considered interpretable, while those that make outputs comprehensible to users are explainable [291].

Explainable AI systems are operational necessities in critical contexts such as financial infrastructures. This requirement stems from multiple imperatives: building user trust, enabling effective debugging, identifying potential biases, and meeting regulatory compliance. From a system management perspective, clear explanations enable operators to assess alert criticality, reduce false positive fatigue, and take appropriate corrective actions [249]. The use of predictive analytics [288] and integration of XAI techniques [92, 344] represent recent trends, moving toward Automated RCA solutions [314, 410, 476], which enable rapid identification of problem causes, thereby accelerating recovery and reducing downtime. The following subsections overview AIOps techniques and XAD methods applicable to time series.

Proactive and Reactive Failure Management

AIOps systems rely on time-series analysis to maintain service availability through proactive failure prediction and reactive anomaly detection. Proactive methods estimate failure probability from current states, enabling preemptive interventions such as workload migration or resource provisioning. Early research focused on hardware reliability using SMART attributes, with Bayesian approaches [150] and SVMs on sliding sensor windows [301] improving detection

rates. Temporal modeling advanced through Hidden Markov and Semi-Markov models on error logs [346, 481], while RNNs demonstrated superior capability in predicting hardware health and remaining useful life [439, 482]. System-level prediction has employed nearest-neighbor methods on application logs [250] and autoregressive integrated moving average (ARIMA) models with fault trees for availability forecasting [59].

When preemptive action is infeasible, reactive detection becomes essential. Early techniques included k-Nearest Neighbors, k-Means clustering, and one-class SVMs. However, the curse of dimensionality causes these methods to perform suboptimally as dimensions increase. Classical statistical methods such as ARIMA and exponentially weighted moving average (EWMA) [339] detect deviations by modeling linear temporal relationships, though they struggle with seasonality and sudden shifts. Extreme value theory, exemplified by SPOT [369], addresses tail distribution anomalies without manual threshold tuning. Supervised frameworks like Opprentice [260] and EGADS [216] apply Random Forests to extracted features, but label scarcity has driven adoption of unsupervised deep learning. Generative models, particularly VAEs, detect anomalies via reconstruction probability: Donut [445] handles missing data in seasonal KPIs, while Bagel [245] incorporates conditional temporal information.

Analyzing metrics independently yields excessive false alarms, necessitating MTS approaches capturing temporal and cross-metric dependencies. PCA-based methods separate normal subspaces from anomalies [214]. Modern architectures include MSCRED [472], using Convolutional LSTMs on system status matrices, and OmniAnomaly [386], employing stochastic recurrent networks. USAD [14] introduces adversarial training for high-dimensional stability, while GNNs such as GDN [95] model topological dependencies to learn causal structures.

Recent advances leverage LLMs for fault diagnosis and RCA, with systems automatically matching incidents to handlers and generating explanatory narratives [68]. Domain-adapted models employ mixture-of-adapter strategies [146], while holistic frameworks decompose complex tasks using network-specialized LLMs [149]. Evaluation frameworks enable assessment through controlled fault injection [67,

361]. Microservice architectures benefit from approaches leveraging multimodal data sources (logs, metrics, and traces) for effective fault localization [477]. Multivariate log-based frameworks aggregate features from distributed nodes for cluster-wide detection [475], while integrated approaches provide precise diagnostic capabilities [419]. AI-driven predictive failure analysis has evolved from statistical methods to sophisticated architectures, including LSTMs, Transformers, and GNNs, enabling proactive maintenance and automated recovery [194, 321].

Explainable Anomaly Detection

Understanding both model functioning and output meaning has become critical as complexity grows and deployment extends to high-stakes contexts [215, 315]. Explanations must be tailored to different audiences: narratives for IT operators may not suffice for AI designers [291]. Four primary explanation strategies exist [457]: feature importance highlights attributes driving decisions [191, 274, 308]; feature values provide rule-based logic such as threshold violations [28, 207, 298, 376]; data point comparison contrasts anomalies with normal instances [209, 247, 374]; and structure analysis examines internal distributions [225, 279, 365]. We categorize techniques by when explainability is introduced: in-model, pre-model, or post-model [249].

A system is interpretable when its internal mechanisms can be inspected directly. Inherently interpretable in-model approaches are transparent by construction, contingent upon simulability, decomposability, and algorithmic transparency [259]. Common techniques include decision trees, linear regression, rule-based systems, and Gaussian processes. Composition-based Decision Trees [208] extract detection rules for univariate time series without manual hyperparameter tuning, though requiring labeled data. Regularized Logistic Regression with Elastic Net [215] handles streaming multivariate time series for short-term predictions but struggles with long-term dependencies. Hybrid models like STRIC [467] balance complexity and transparency by combining interpretable linear filters with deep temporal networks.

When systems function as black boxes, explanations can still be generated through separate explanation systems [32]. Pre-model (ante-hoc) techniques optimize feature selection or representation before detection, operating on the premise that investigation effort scales with feature count [368]. Symbolic representations using Piecewise Aggregate Approximation [330] enhance interpretability through dimensionality reduction and enable human-in-the-loop analysis, though limited to symbolic-based detectors. Deviation convolution-based models [352] learn characteristic shapes of normal patterns, facilitating visualization but potentially missing less frequent patterns. Dimensionality reduction techniques like PCA and independent component analysis (ICA) improve computational efficiency but sacrifice interpretability by transforming features into components no longer clearly discernible, highlighting the trade-off between complexity reduction and transparency.

Post-model (post-hoc) techniques generate explanations after training and inference, operating model-agnostically by correlating inputs and outputs [274]. InterFusion [248] employs Markov chain Monte Carlo (MCMC) sampling to identify anomalous metrics in streaming multivariate time series, yet lacks adaptability to evolving distributions. DAEMON [66] leverages reconstruction errors in Adversarial AEs to pinpoint contributing features. Attention mechanisms in LSTMs [51] provide feature importance and relationship mapping. Modified Kernel Shapley additive explanations (SHAP) [137] adapts Shapley values for GRU-based AEs to identify contributing time windows, assuming feature independence. Feature value-based methods convert decisions into human-readable logic: TSXplain [298] connects statistical features with deep network outputs to generate natural language, while EXAD [376] approximates deep models via decision trees to extract Disjunctive Normal Form rules. Visualization strategies are effective for sequential data. ARIMA models with Virtual Reality tools [8] visualize residuals against predictions. The Automata Violation Analyzer [18] compares observed sequences with those expected from reference automaton models. Spatiotemporal visualization frameworks [286] correlate traffic anomalies with time and location using Z-scores of kernel density values.

For opaque models, integrating domain-specific constraints into the detection pipeline improves interpretability, termed Informed Machine Learning [414]. Business rules combined with surrogate models explain fuel consumption anomalies [27], Dynamic Bayesian Networks encode physical constraints for autonomous driving [373], and hierarchical clustering incorporates expert knowledge for network traffic analysis [296]. These strategies enable validation against established principles, enhancing confidence in detected anomalies.

Part II

NOVEL APPROACHES

Time is not a line—it's a spiral.
Generate sets, not sequences.

4

HELICAL TIME ENCODING FOR SEASONAL TIME SERIES

Handcrafted feature engineering for time series is labor-intensive, requires domain expertise, and yields rigid representations that often fail to capture complex temporal dynamics [65]. While time-encoding methods such as Time2Vec [192] provide expressive temporal embeddings that improve model performance, they typically lack interpretability or invertibility. This chapter proposes HTE, a model-agnostic geometric encoding that maps onto a helix in a higher-dimensional space (RQ1). The encoding maps temporal relationships by positioning observations in a space where seasonality is embedded in the geometry, ensuring orthogonality between cyclic dynamics and linear progression. This structure ensures topological alignment: seasonally equivalent observations remain geometrically proximal, and invertibility guarantees the deterministic recovery of the temporal phase.

4.1 HELICAL TIME ENCODING REPRESENTATION

Helical Time Encoding (HTE) [324] represents a seasonal time series as a trajectory on a helix to explicitly encode recurring patterns and temporal progression in geometric space.

Given a time series $\{\mathbf{y}_t\}_{t=0}^{T}$ with $\mathbf{y}_t \in \mathbb{R}^d$, each timestamp-observation pair $(t, \mathbf{y}_t)$ is mapped to a point in $(d+3)$-dimensional space as follows:

$$\mathbf{h}_t = [\mathbf{y}_t, x(t), y(t), z(t)]. \tag{4.1}$$

The components $x(t), y(t), z(t)$ of the vector $\mathbf{h}_t$ represent the parametric coordinates of a helix defined as:

$$x(t) = \cos\left(\frac{2\pi t}{\lambda} + \varphi\right), \tag{4.2}$$

$$y(t) = \sin\left(\frac{2\pi t}{\lambda} + \varphi\right), \tag{4.3}$$

$$z(t) = p(t) + \vartheta. \tag{4.4}$$

The constant λ determines which seasonal pattern is mapped to one complete helix rotation, defining the base resolution of the temporal encoding. The coordinates $x(t)$ and $y(t)$ establish temporal proximity relationships, ensuring that times t and $t + k\lambda$ (for integer k) have identical (x, y) coordinates and differ only in the z-component. The progression function $p(t)$ tracks evolution across different helix rotations, capturing both long-term trends and cycle variations. The parameters φ and ϑ represent the phase and vertical offsets, respectively.

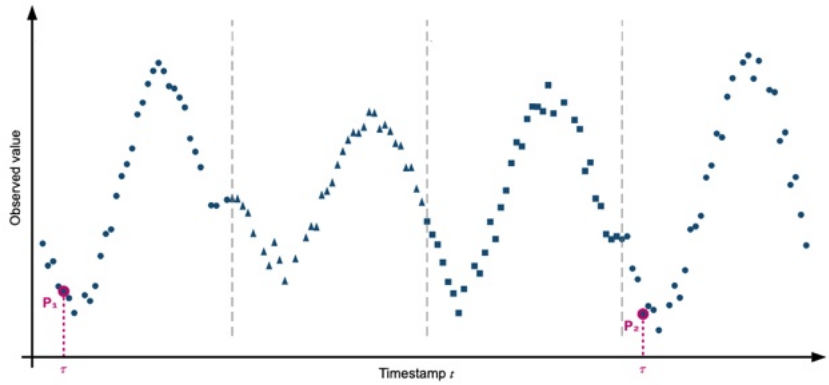
Figure 4.1 illustrates HTE for a univariate seasonal time series. In the Cartesian representation (Figure 4.1a), observations P_1 and P_2 have no explicit temporal relationship despite being located at the same relative point within a seasonal pattern. The helical representation (Figure 4.1b) vertically aligns these observations at the same helical phase.

4.2 PROPERTIES OF HELICAL TIME ENCODING

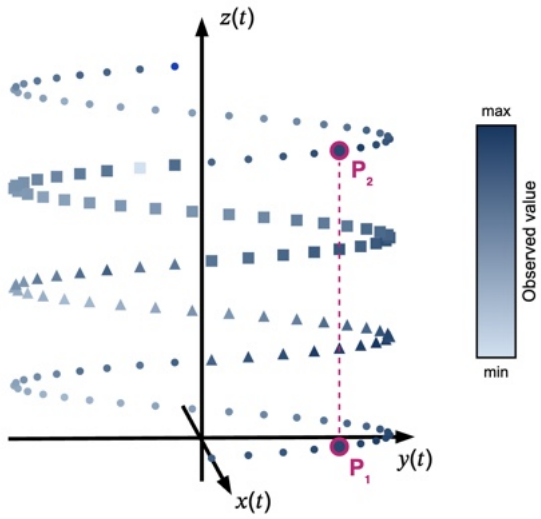
HTE exhibits five properties that characterize its geometric structure and define how temporal information is distributed across spatial dimensions:

P1. *Seasonal Periodicity.* The (x, y) coordinates provide a periodic mapping where observations separated by integer multiples of λ share identical circular coordinates. This creates a topological alignment where seasonally equivalent observations are mapped to the same position on the transverse plane, independent of their absolute temporal distance.

P2. *Temporal Progression.* The z-component, controlled by the progression function $p(t)$, captures the irreversible flow of time and inter-seasonal dynamics. This dimension enables distinct seasonal cycles sharing the same periodic phase to be differentiated, preserving global temporal order.



(a) Observations P_1 and P_2 occupy the same relative position within their respective seasonal patterns but lack an explicit relationship in this representation, with timestamps t on the x-axis and observed values on the y-axis. Dotted vertical lines partition the series into seasonal periods.



(b) In the helical representation, observations P_1 and P_2 are explicitly aligned at the same helical phase. Observation values are shown via heatmap coloring.

Figure 4.1: Representation of a univariate seasonal time series with HTE. Marker shapes indicate different patterns in the seasonal time series.

P3. *Multi-scale Proximity.* Euclidean distances in the helical space represent a composite similarity metric. Observations that are close in both seasonal phase and absolute time maintain small geometric distances, while seasonally similar but temporally distant observations remain proximal in the (x, y) subspace but are separated along the z-axis.

P4. *Component Separability.* The encoding decomposes temporal information into orthogonal periodic (x, y) and progressive (z) components. This separation allows cyclic features (seasonality) and linear features (trend or evolution) to be processed as distinct information channels.

P5. *Phase Recovery.* The projection of the helix onto the two-dimensional (x, y) plane enables deterministic retrieval of the seasonal phase. While the absolute timestamp requires the z-component, the relative position of an observation within its seasonal cycle can be reconstructed using only the (x, y) coordinates and inverse trigonometric functions.

4.3 INTEGRATION WITH DEEP LEARNING ARCHITECTURES

Integrating HTE into deep learning frameworks enhances the learning of temporal dynamics. Depending on the architecture, the encoding serves either as auxiliary context for sequential models or as a geometric structure for generative approaches.

4.3.1 *Augmenting Sequential Models*

In standard regression-based forecasting models, such as RNNs, CNNs, or MLPs, HTE is concatenated to observed variables to enrich the input feature space. Here, the encoding acts as a strong inductive bias. By explicitly providing the cyclic coordinates (x, y), the network avoids learning periodic functions from raw indices, enabling it to focus on modeling residuals and local anomalies. Furthermore, the z-component provides a normalized reference for trend progression, resolving the issue of unbounded temporal values that often destabi-

lize gradient descent in long-term series. In this context, $\mathbf{h}_t$ anchors the hidden states of the model to a temporal context, facilitating the capture of long-range dependencies.

4.3.2 *Enabling Generative Paradigms*

Integrating HTE into generative models alters the modeling paradigm. While sequential models rely on the conditional probability $P(\mathbf{y}_t|\mathbf{y}_{t-k},\ldots,\mathbf{y}_{t-1})$, the properties of HTE allow modeling the joint distribution of observations and time $P(\mathbf{y},\mathbf{h})$.

By mapping the temporal domain onto a helical curve in higher-dimensional space, the forecasting problem is reformulated from sequence extrapolation to conditional generation on geometric coordinates. HTE transforms the time series into a set of independent data points distributed in $(d+3)$-dimensional space. Consequently, if a generative model learns the mapping between the helical coordinates and the observation distribution, it can synthesize data points based solely on their location on the helix. This property is fundamental to the framework presented in Chapter 5, as it enables the generation of observations without historical input windows at inference time.

5 GENERATION WITH THE TIMESTAMP TRICK

Generative models for time series often rely on sequential processing, requiring sliding windows of historical data for synthesis. This chapter presents Generation with the Timestamp Trick (GenTT), a framework that reformulates time series synthesis as a conditional generation task by leveraging explicit time encodings (RQ2). While the framework operates with any encoding that explicitly represents seasonal patterns and is invertible, we present it using HTE. The framework trains models to learn the joint distribution of observations and temporal representations, constraining the generator to output temporal indices alongside target variables. This enables the reconstruction of sequential order from independently generated samples, shifting from sequential extrapolation to direct temporal conditioning and thereby allowing history-free inference, where future observations are generated solely from their temporal position without requiring historical context.

5.1 TIME SERIES PREPROCESSING

Consider a seasonal time series with d variables observed at regular time intervals, assuming at least one known seasonal pattern of length λ (e.g., daily, weekly, or annual). We denote this time series as:

$$\{\mathbf{y}_t\}_{t=0}^{T} = \left\{ \begin{bmatrix} y_{1,0} \\ y_{2,0} \\ \vdots \\ y_{d,0} \end{bmatrix}, \begin{bmatrix} y_{1,1} \\ y_{2,1} \\ \vdots \\ y_{d,1} \end{bmatrix}, \ldots, \begin{bmatrix} y_{1,T} \\ y_{2,T} \\ \vdots \\ y_{d,T} \end{bmatrix} \right\}, \tag{5.1}$$

where T is the total length of the time series, $y_{i,t}$ is the observation of the i-th variable at time t, and $\mathbf{y}_t$ is a $d \times 1$ vector of observations at time t. If $d = 1$, the time series is univariate; otherwise, it is multivariate.

To generate temporally situated data, the components $x(t)$ and $y(t)$ of the vector $\mathbf{h}_t$ (see Eq. 4.1) must be learned and generated alongside the target observations. Leveraging the component separability property P4, we treat these coordinates as target features. The training set is thus represented by the concatenation of these two geometric features with the observed variables:

$$\left\{ \begin{bmatrix} \mathbf{y}_t \\ x(t) \\ y(t) \end{bmatrix} \right\}_{t=0}^{T} = \left\{ \begin{bmatrix} y_{1,0} \\ \vdots \\ y_{d,0} \\ x(0) \\ y(0) \end{bmatrix}, \begin{bmatrix} y_{1,1} \\ \vdots \\ y_{d,1} \\ x(1) \\ y(1) \end{bmatrix}, \ldots, \begin{bmatrix} y_{1,T} \\ \vdots \\ y_{d,T} \\ x(T) \\ y(T) \end{bmatrix} \right\}. \tag{5.2}$$

5.2 CONDITIONAL GENERATION

In the GenTT framework, the helical vector $\mathbf{h}_t$ from Eq. 4.1 acts as the structural backbone. While $x(t)$ and $y(t)$ are generated, the remaining component $z(t)$ serves as a condition. Conditioning can be extended with a component $c(t) : \mathbb{Z} \to \mathcal{C}$, which enables modeling higher-order seasonal patterns by partitioning data based on distinct temporal characteristics (e.g., day-of-week or holidays). This allows the model to capture structural variations within the primary seasonal pattern. The discrete conditions $\mathcal{C} = \{c_1, c_2, \ldots, c_{|\mathcal{C}|}\}$ can be specified based on domain knowledge.

During training and inference, components $c(t)$ and $z(t)$ serve as conditioning inputs. The condition $c(t)$ enhances the seasonal periodicity property P1 by enabling explicit modeling of different pattern categories, while $z(t)$ exploits the temporal progression property P2 to capture the evolution of the time series.

The helical representation provides a geometric structure that makes temporal proximity explicit (property P3), allowing the generative model to learn the patterns of the underlying probability distribution. Recurrent patterns emerge naturally from the interaction between the helical coordinate system and the learning process. The model observes repeated associations between helix coordinates and value distributions during training, learning to generate samples

that respect both the expected patterns for given coordinates and the temporal structure.

In the training phase, observations are treated as independent samples from a distribution rather than sequential data. Generated data points constitute vectors in $\mathbb{R}^{d+2}$ distributed across the seasonal period λ. The coordinates $\hat{x}$ and $\hat{y}$, generated alongside target features, serve as temporal anchors for reordering independently generated observations. Once trained, the generative model can synthesize new observations by conditioning solely on timestamp-derived parameters for future time points, eliminating the need to access recent observed values.

5.3 THE TIMESTAMP TRICK

Since generated samples are produced independently, they lack inherent temporal ordering. The timestamp trick constrains the generative model to output the x and y coordinates alongside target observations, exploiting the phase recovery property P5 of HTE. This allows the reconstruction of temporal indices from the generated coordinates $(\hat{x}, \hat{y})$, enabling the sequential arrangement of the samples.

When projecting the helix onto the xy-plane, we observe discrete points distributed along a circle's circumference, as shown in Figure 5.1. Due to the surjective mapping between the helix and the circumference, the coordinates identifying a point P correspond to the observation's position within the seasonal pattern.

Given generated coordinates $P = (\hat{x}, \hat{y})$, we first recover the angular position using the arctan 2 function, which provides the angle between the positive x-axis and point P:

$$\alpha = \arctan 2(\hat{y}, \hat{x}) \bmod 2\pi. \tag{5.3}$$

The modular operation ensures the angle falls within the valid range $[0, 2\pi)$.

The corresponding index τ within the seasonal pattern is obtained by mapping the continuous angular position to discrete time indices:

$$\tau = \text{round}\left(\alpha \cdot \frac{\lambda}{2\pi}\right) \bmod \lambda. \tag{5.4}$$

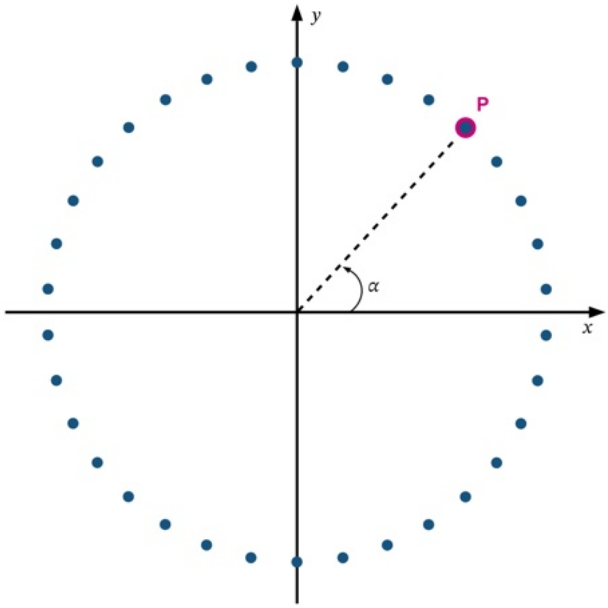


Figure 5.1: Projection of the helix onto the xy-plane. The angle between the positive x-axis and point P enables the computation of the index for sample reordering.

The modular operation constrains the time index to $[0, \lambda)$. For example, if one helix rotation represents a daily pattern in seconds, the index τ corresponds to the number of seconds from midnight.

After obtaining τ for each observation, we sort the generated samples by their temporal indices to establish the ordering: $\{(\hat{\mathbf{y}}_{(1)}, \tau_{(1)}), \ldots, (\hat{\mathbf{y}}_{(m)}, \tau_{(m)})\}$ where $\tau_{(1)} \leq \tau_{(2)} \leq \ldots \leq \tau_{(m)}$.

The final time series is obtained by partitioning the temporal domain $[0, \lambda)$ into b uniform bins and aggregating samples within each bin using functions such as the mean or median.

5.4 SAMPLING STRATEGY AND TEMPORAL COVERAGE

Since samples are generated randomly, the reconstructed time series may contain temporal gaps. To ensure complete temporal coverage, all temporal bins must receive at least one generated sample.

The number of bins b depends on the sampling frequency ν:

$$b = \frac{\lambda}{\nu} \tag{5.5}$$

Assuming the generator produces temporally uniform samples, justified by the uniform distribution of training data, each sample has an equal probability of falling into any bin. The probability $P(m, b)$ that at least one sample falls into each bin is:

$$P(m, b) = \frac{b!}{b^m} \left\{ {m \atop b} \right\} \tag{5.6}$$

where $\left\{ {m \atop b} \right\}$ denotes the Stirling number of the second kind.

For practical implementation, we set $m = b \cdot s$, where s is a smoothing factor controlling the trade-off between computational overhead and prediction smoothness. For example, with $b = 72$ and $s = 15$, we achieve $P(m, b) \approx 0.99998$, making empty bins negligible. When multiple samples fall into the same bin, aggregation via the mean reduces random fluctuations, resulting in smoother predictions.

5.5 FRAMEWORK APPLICATIONS

The GenTT framework can be applied to various seasonal time series tasks. This section describes its extension to forecasting and anomaly detection, demonstrating the end-to-end workflow. For implementation details, refer to Appendix a.

5.5.1 *Forecasting*

GenTT differs from traditional deep learning forecasting paradigms in that historical observations are used only during training. At inference time, the latent vector is conditioned only on timestamp-derived features $c(t)$ and $z(t)$, as illustrated in Figure 5.2.

This approach modifies standard machine learning pipelines (Figure 5.3). Data preparation aggregates raw time series into temporal buckets to ensure uniform sampling. During preprocessing, HTE and day-of-week labeling extract timestamp-derived features. Unlike

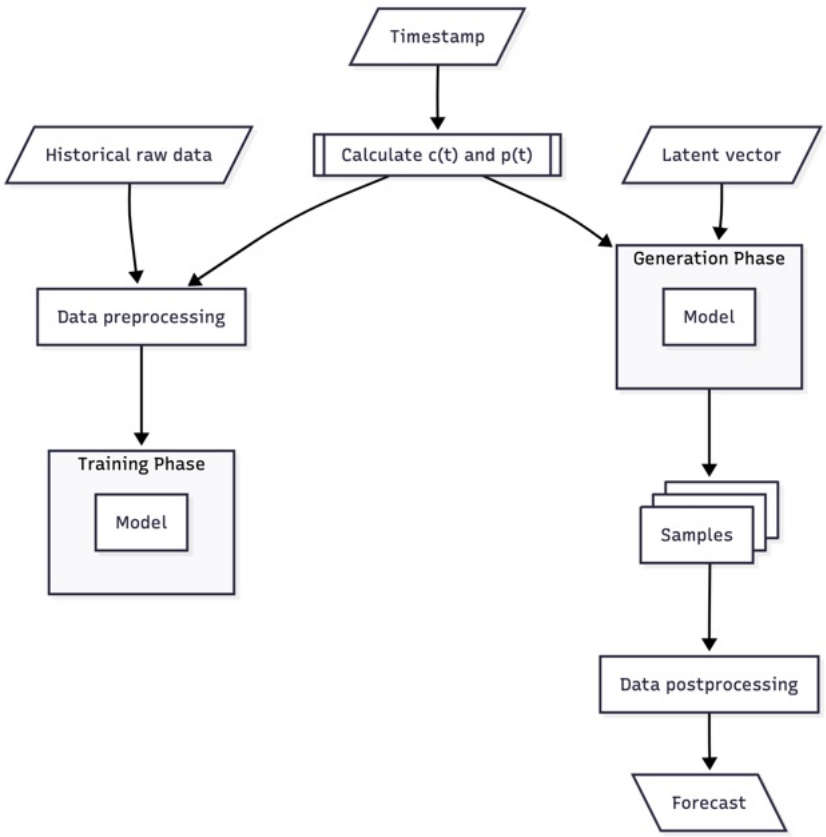
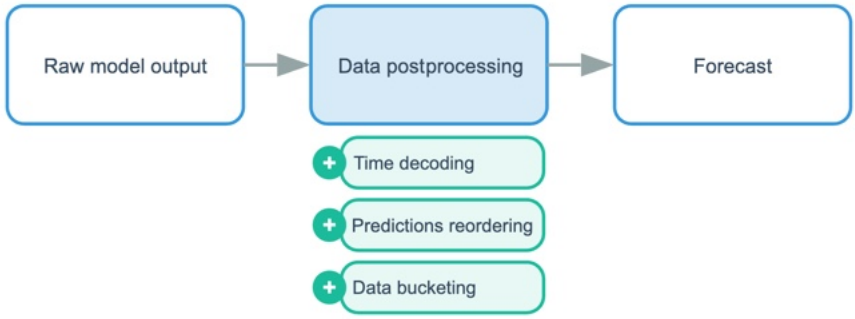


Figure 5.2: Overview of conditional forecasting with GenTT. During training, the model learns the distribution of observations for given timestamps. At inference time, forecasts are generated using conditional labels derived from timestamps.

(a) Preprocessing pipeline: time series timestamps are encoded as two additional artificial features.



(b) Postprocessing pipeline: temporal indices τ are decoded from predictions, observations are reordered, and aggregated to obtain the final forecast.

Figure 5.3: Data processing pipeline modifications required by the GenTT framework.

sliding window methods that segment time series into overlapping subsequences, observations are processed individually in batches without temporal windowing.

The model is trained on individual observations conditioned on their timestamp-derived features, learning the seasonal distribution without sequential context. During inference, the model synthesizes predictions as individual observations. Postprocessing reconstructs the temporal sequence by applying the inverse helical mapping to reorder generated samples, then aggregates individual predictions to produce the final forecast.

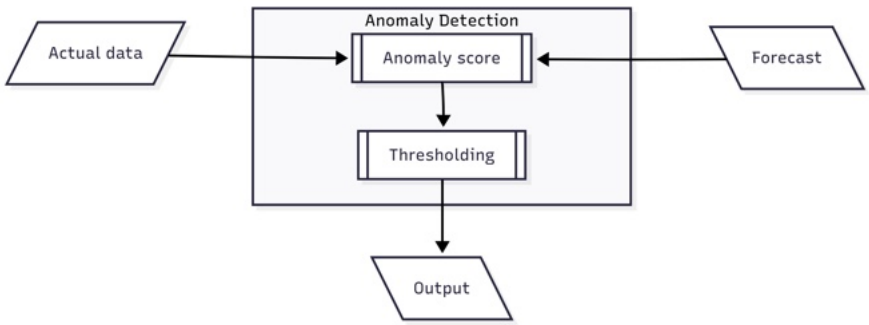


Figure 5.4: Prediction-based anomaly detection with the GenTT framework. Once forecasts are obtained, anomaly detection is performed using residuals between predictions and observed values as anomaly scores.

5.5.2 *Anomaly Detection*

For anomaly detection, GenTT generates a seasonal baseline representing expected behavior, forming a prediction-based anomaly detection approach (Figure 5.4). The workflow follows the same data preparation and training procedures as forecasting.

By conditioning the generator on future timestamps, the model produces the probability distribution of expected values at those points based on learned seasonal structure rather than potentially anomalous input values. As the model learns dominant patterns of the data distribution during training, it naturally filters out transient outliers during generation.

After synthesis and temporal reordering, anomaly scoring computes a distance metric between the actual observation $\mathbf{y}_t$ and the generated baseline $\hat{\mathbf{y}}_t$. Large distances indicate divergence between observed and expected behavior. This approach eliminates the need for lookback buffers, enabling real-time monitoring with reduced latency.

Part III

EXPERIMENTAL VALIDATION

Structure reveals both path and error.
High impact, low footprint.

6 TIME SERIES FORECASTING

Forecasting time series over extended horizons is constrained by sliding window architectures and the accumulation of autoregressive errors. This chapter instantiates the GenTT framework using HTE and cGANs to reformulate seasonal time series forecasting as conditional generation on a helical representation. The geometric encoding makes seasonal patterns explicit, allowing models to learn position-value relationships rather than sequential dependencies. The framework converts independent samples into coherent forecasts and supports the direct prediction of entire seasonal periods without requiring recent observations at inference time. This chapter addresses RQ3 by evaluating HTE in two forecasting paradigms: as an auxiliary feature augmenting traditional sequential architectures (RQ3a), and as the core temporal representation within the GenTT framework (RQ3b).

6.1 INTRODUCTION

Time series forecasting supports decision-making across multiple domains, including weather systems, energy grids, public health monitoring, and urban mobility. For instance, in Intelligent Transportation Systems for smart cities, demand forecasting informs vehicle allocation and routing decisions that reduce congestion and lower carbon emissions [182, 251, 381], contributing to environmental sustainability [230]. Long-term forecasting, extending from days to years depending on system dynamics [78, 334], enables proactive resource allocation and early warning systems [432]. Classical statistical methods like ARIMA [45] and machine learning approaches like XGBoost [63] remain effective when distributional assumptions hold and on simpler univariate problems [206, 283]. However, they

struggle with nonlinear relationships, missing data, and extended horizons as system complexity scales.

Deep learning approaches have emerged to address these limitations through flexible function approximation [160, 354] and native support for multivariate inputs [204, 213, 238, 255, 313, 349, 486]. Transformer architectures have particularly advanced the field by capturing long-range dependencies [52, 97]. However, standard machine learning forecasting relies predominantly on sliding windows, where historical observations are required to predict future values as the window shifts through the series. This approach requires extensive hyperparameter tuning of the window size and faces efficiency constraints when modeling long-term dependencies, as the context window creates a trade-off between information retention and computational cost. Furthermore, recursive prediction strategies in these architectures often lead to error accumulation over long horizons.

To address these structural limitations, we propose the HTE-cGANs model, an instantiation of the GenTT framework using cGANs with HTE, which reformulates seasonal forecasting as conditional generation. Rather than learning patterns implicitly from sequences, our approach explicitly represents periodicity and temporal progression through geometric coordinates in a helical space. This representation enables the generation of future observations derived solely from timestamp features, eliminating the strict dependency on recent historical values during inference. By treating training data as individual observations conditional on their geometric position rather than as sequential windows, the encoding removes the need for window size optimization while enabling inference that generates complete seasonal periods in a single step.

We validate our approach on taxi demand prediction [101, 193, 240, 264, 293, 403, 454, 455, 465] and influenza forecasting, demonstrating competitive performance against established baselines, particularly in scenarios where recent observations are unavailable. Section 6.2 describes the experimental setup, Section 6.3 presents the results, and Section 6.4 discusses findings, implications, and limitations. Finally, Section 6.5 examines threats to validity.

6.2 EXPERIMENTAL SETUP

We formalize the long-term forecasting problem as predicting the next m seasonal periods $\{\hat{\mathbf{y}}_t\}_{t=T+1}^{T+m\lambda}$ given a historical time series $\{\mathbf{y}_t\}_{t=1}^{T}$, where λ denotes the seasonal period length. At inference time, predictions must be generated without recursive access to previous predictions or future observations $(\mathbf{y}_{T+1}, \mathbf{y}_{T+2}, \ldots)$, whereas during training, the model has access to the complete historical dataset.

This formulation differs from practical implementations with sliding windows, which typically require recent observed values during prediction to prevent error accumulation through autoregressive chains. Our approach eliminates the runtime requirement for new observations: no sliding windows, autoregressive context, or rolling origin updates are needed when generating forecasts. The model learns the joint distribution of observations and their temporal positions during training, then generates future values conditioned only on timestamp-derived features.

We evaluate the proposed forecasting approach on two real-world seasonal time series: Influenza-like Illness incidence and NYC Taxi Demand.

6.2.1 *Datasets*

Our experiments consider two distinct seasonal forecasting problems representing different temporal scales and application domains. Additionally, we created a synthetic dataset to conduct ablation studies aimed at evaluating the model's ability to capture multiple seasonalities and trend.

INFLUENZA-LIKE ILLNESS (ILI). The ILI dataset contains surveillance data from the United States Centers for Disease Control and Prevention [409], recording the incidence of symptoms consistent with influenza. These data enable the monitoring of seasonal flu activity, supporting timely public health responses and resource planning. We selected the period from 2002 to 2021, which includes weekly

observations exhibiting annual seasonality and an upward trend. Following previous works [432], we used a 7:1:2 chronological split. However, to compare generative and autoregressive models under equal information access conditions and eliminate advantages stemming from more recent ground truth data, only the most recent 95 observations from the past two years are considered in the test set for evaluation metric calculation.

NYC TAXI DEMAND (NYC TLC). Taxi demand forecasting represents a smart city application where accurate demand prediction enables optimal vehicle allocation [327, 396, 405, 429, 447]. Yellow taxi trip records from Manhattan (January–March 2016, 91 days total) were used [79]. Following prior works [449], data are aggregated into 20-minute intervals. The time series exhibits both daily and weekly seasonality. The dataset is split chronologically: 73 days for training, 18 days for testing. To evaluate model robustness, two additional NYC TLC training set variants were created: TRN-1, containing the first 43 days before the test set, and TRN-2, derived from TRN-1 with 2% randomly distributed outliers introduced.

SYNTHETIC TIME SERIES (SYNTHTS). This dataset consists of a synthetic univariate time series covering two years with a regular 20-minute sampling frequency. The dataset exhibits clear intra-daily seasonality, simulating patterns that repeat on a daily basis. Weekly seasonality is also introduced, with higher values during weekends compared to weekdays. The dataset incorporates a linear increasing trend over time, representing a general value increase throughout the observation period. To enhance realism, random noise modeled as a normal random variable was added to simulate unpredictable fluctuations. Figure 6.1 shows the complete time series.

6.2.2 *Baseline Methods*

To evaluate HTE as an additional feature for baseline methods, we implemented standard versions of MLPs, CNNs, and LSTMs networks using TensorFlow. Implementations assess the impact of the

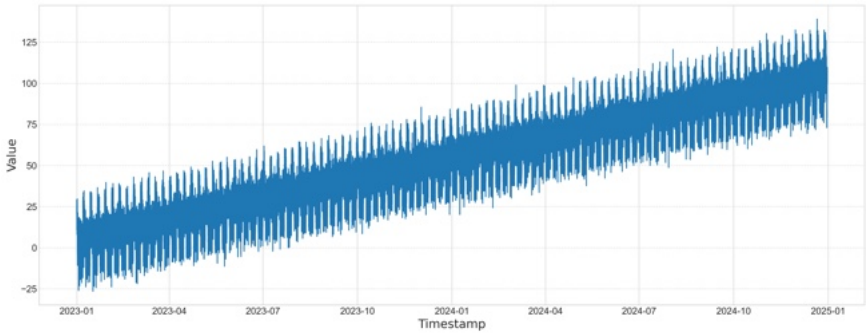


Figure 6.1: Synthetic time series (SynthTS) used in the ablation study.

proposed representation as an additional feature alongside training data. We used default parameters for comparison. The MLP uses a dense layer with 512 units and ReLU activation; the CNN employs a one-dimensional convolutional layer with 256 filters; and the LSTM consists of a single layer with 32 units. All models were trained with the Adam optimizer, mean squared error loss, and early stopping.

We then validate GenTT implementations using GANs with HTE, comparing them against state-of-the-art methods on NYC TLC and ILI datasets in two prediction modes.

For experiments on the NYC TLC dataset, benchmarking was performed against nine state-of-the-art methods, including traditional statistical approaches and machine learning techniques: (1) Historical Average, (2) ARIMA [45], (3) Linear Regression, (4) Ridge Regression [166], (5) Random Forest [253], (6) eXtreme Gradient Boosting (XGBoost) [63], (7) MLP, (8) Origin-Destination-based Temporal Graph Attention Network (OD-TGAT) [449], and (9) ST-ResNet [474]. These models generated forecasts using the most recent observed values at prediction time, leveraging up-to-date data to inform their predictions.

For experiments on the ILI dataset, a different evaluation strategy was adopted: no baseline model had access to actual observed data beyond the initial window. All forecasts were produced in an iterative (autoregressive) manner, using only previously predicted values as input for subsequent predictions. This setting simulates long-term forecasting scenarios where ground truth data are unavailable after the

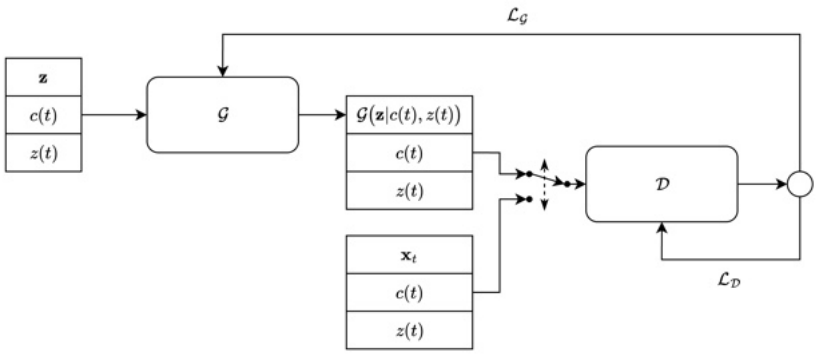


Figure 6.2: HTE-cGANs architecture showing generator $\mathcal{G}$ synthesizing data points from latent vector $\mathbf{z}$, conditioned on $c(t)$ and $z(t)$, and discriminator $\mathcal{D}$ evaluating real and generated samples.

initial window. Comparisons were made against Autoformer [432], Informer [486], Reformer [204], and standard Transformer, using the implementations provided by the Autoformer authors [390].

6.2.3 *Evaluation Metrics*

Standard forecasting metrics MAE and RMSE were used to enable direct comparison with existing literature [432, 449]. For the definition of these metrics, refer to Section 2.5.1.

6.2.4 *Implementation details*

We instantiate the GenTT framework by combining conditional GANs [141] with HTE, yielding the HTE-cGANs model. The architecture (Figure 6.2) comprises a generator $\mathcal{G}$ and discriminator $\mathcal{D}$. The generator synthesizes the joint distribution of observations and their temporal coordinates by taking a latent noise vector $\mathbf{z} \sim \mathcal{N}(0, I)$ and temporal conditions $c(t)$ and $z(t)$ as input, then outputting both target variables and periodic coordinates required for the timestamp trick. The discriminator evaluates sample authenticity, distinguishing real observations from generated ones.

Component	Parameter	Value
Conditioning	Labelling function $c(t)$	Day-of-week
	Progression function $p(t)$	–
Generator	Units / Layers	[64, 64]
	Activation function	ReLU
	Latent space dimensions	5
	Optimizer	Adam
	Learning rate	0.0003
	β_1, β_2, ϵ	0.5, 0.999, 1e-07
Discriminator	Units / Layers	[128, 64, 32]
	Activation function	ReLU
	Dropout	[-, 0.3, 0.3]
	Optimizer	Adam
	Learning rate	0.0005
	β_1, β_2, ϵ	0.9, 0.999, 1e-07

Table 6.1: HTE-cGANs hyperparameters and training configuration for the NYC TLC dataset.

The generator concatenates the latent vector with $c(t)$ and $z(t)$ before feeding them through MLP layers. Specific architectural hyperparameters, including the number of dense layers, are dataset-dependent as detailed in Tables 6.1 and 6.2.

We exploit the modularity of HTE by using only the components necessary to capture each dataset's temporal structure. For NYC TLC data, which exhibits daily and weekly patterns, we model the daily cycle on the helix while allowing the network to capture intra-day variations. Since this series lacks clear trends, we omit the z-component of the helix, simplifying the architecture. For ILI data, annual seasonality maps to one complete helical rotation. We capture temporal progression through:

$$p(t) = \frac{t - t_0}{\mathrm{Y}} \tag{6.1}$$

Component	Parameter	Value
Conditioning	Labelling function $c(t)$	–
	Progression function $p(t)$	Yearly progression
Generator	Units / Layers	[128, 256, 512]
	Activation function	ReLU
	Latent space dimensions	4
	Optimizer	Adam
	Learning rate	8.542e-03
	β_1, β_2, ϵ	0.847, 0.999, 1e-07
Discriminator	Units / Layers	[128, 64, 32]
	Activation function	ReLU
	Dropout	[-, 0.3, 0.3]
	Optimizer	Adam
	Learning rate	1.369e-05
	β_1, β_2, ϵ	0.915, 0.999, 1e-07

Table 6.2: HTE-cGANs hyperparameters and training configuration for the ILI dataset.

where t_0 denotes the reference timestamp (first day of the training year) and Y provides yearly normalization. This dataset requires no additional conditioning on $c(t)$.

6.3 RESULTS

This section evaluates HTE's contribution to forecasting accuracy across two paradigms. We first assess how this time representation enhances traditional sequential architectures when used as an auxiliary input feature (RQ3a). We then examine the HTE-cGANs model within the GenTT framework, comparing its performance against established baselines (RQ3b). Following the accuracy evaluation, we systematically investigate the contribution of individual encoding components through ablation studies, examine robustness under de-

graded training conditions, analyze hyperparameter sensitivity, and computational performance.

6.3.1 *Helical Encoding as Sequential Model Augmentation*

We first investigate if HTE improves forecasting performance when incorporated as an additional feature in MLP, CNN, and LSTM deep learning architectures. Table 6.3 presents results for models trained on univariate time series with and without HTE on the reduced TRN-1 dataset, across multiple window configurations. Error reduction is consistent across all architectures when HTE is included. For CNN models, MAE improvements range from 29.7% to 53.8%, while LSTM architectures achieve reductions between 33.7% and 38.3%. MLP models exhibit more variable behavior, with improvements ranging from 22.8% to 60.9% depending on configuration. Results also indicate that window hyperparameter selection substantially affects performance. Multi-step forecasting horizons (72 steps) frequently yield lower error rates than single-step predictions across configurations, suggesting that longer horizons reduce error accumulation.

6.3.2 *Conditional Generation with HTE-cGANs*

This section evaluates HTE integration within the GenTT framework through the HTE-cGANs model. We compare generative forecasting performance against established baselines, followed by systematic analysis of model components, robustness, hyperparameter sensitivity, and computational requirements.

6.3.2.1 *Forecast Accuracy Comparison*

Table 6.4 shows results on the ILI dataset where baseline methods perform long-term predictions autoregressively without access to subsequent ground truth observations. HTE-cGANs achieves a 42.2% MAE reduction (from 1.1378 to 0.6582) and 35.3% RMSE reduction (from 1.3063 to 0.8453) compared to the best-performing baseline, Autoformer.

				w/o HTE		with HTE	
Model	Lags	Horiz.	CNN ker.	MAE	RMSE	MAE	RMSE
CNN	504	1	3	0.8748	1.1263	0.5903	0.8151
	504	72	72	0.7210	0.9795	0.3639	0.4844
	72	1	3	1.2089	1.4693	0.8499	1.0884
	72	72	72	1.1229	1.4096	0.5183	0.6837
LSTM	504	1	NA	0.8092	1.0681	0.4989	0.7093
	504	72	NA	0.6237	0.7993	0.3757	0.4921
	72	1	NA	1.1461	1.4181	0.7051	0.9331
	72	72	NA	**0.5861**	**0.7751**	0.3885	0.5309
MLP	504	1	NA	0.8939	1.0717	0.6903	0.9334
	504	72	NA	0.7973	0.9832	**0.3325**	**0.4552**
	72	1	NA	1.7917	2.0255	1.2763	1.5297
	72	72	NA	0.9467	1.1119	0.3704	0.4945

Table 6.3: Impact of HTE on traditional forecasting architectures with varying window widths (lags) and forecasting horizons on the NYC TLC TRN-1 dataset. The best values are in bold.

Table 6.5 presents taxi demand forecasting results. HTE-cGANs achieves a 70.7% RMSE reduction compared to OD-TGAT (from 0.7926 to 0.2322), though MAE remains comparable (0.1696 vs 0.1602). The substantial reduction in RMSE relative to the modest change in MAE indicates that large individual errors have decreased, as RMSE is more sensitive to large deviations than MAE. Unlike baseline models, HTE-cGANs generates predictions directly from timestamp features without requiring recent observations at inference time.

6.3.2.2 *Ablation Study*

To isolate the contribution of individual HTE components, we conducted systematic ablation experiments on a synthetic dataset exhibiting three temporal characteristics: monotonic trend, weekly seasonality, and alternating daily patterns for weekdays versus weekends. The full HTE-cGANs configuration incorporates helical coordinates $x(t), y(t)$ for temporal ordering via the timestamp trick, conditioning

Model	I	O	MAE	RMSE
Transformer	36	24	1.6732	1.9172
	36	36	1.7506	1.9852
	36	48	1.6715	1.8930
	36	60	1.7547	1.9650
Informer	36	24	2.1226	2.3729
	36	36	2.0426	2.2770
	36	48	1.9872	2.2172
	36	60	2.0410	2.2720
Reformer	36	24	1.7388	1.9540
	36	36	1.6511	1.8625
	36	48	1.8247	2.0202
	36	60	1.8352	2.0173
Autoformer	36	24	1.1378	1.3063
	36	36	1.2966	1.5117
	36	48	1.3256	1.5198
	36	60	1.3218	1.5149
HTE-cGANs	-	52	**0.6582**	**0.8453**

Table 6.4: HTE-cGANs forecasting accuracy comparison with sliding-window transformer models on the ILI dataset with various prediction lengths $O \in \{24, 36, 48, 60\}$ and a fixed input length I of 36 observations. Baseline models operate in autoregressive mode without access to ground truth during inference. The best values are in bold.

Model	MAE	RMSE
Historical Average	0.2267	1.2214
ARIMA	0.2033	0.9927
Linear Regression	0.1763	0.9027
MLP	0.1801	0.9072
Ridge Regression	0.1793	0.8862
Random Forest	0.1922	0.9191
XGBoost	0.1848	0.8915
ST-ResNet	0.1628	0.8374
OD-TGAT	**0.1602**	0.7926
HTE-cGANs	0.1696	**0.2322**

Table 6.5: Forecasting accuracy comparison on the NYC TLC dataset. Baseline models use sliding windows of recent observations for prediction. The best values are in bold.

variable $z(t)$ capturing cycle progression and trend, and conditioning variable $c(t)$ distinguishing day types.

Table 6.6 shows that each component removal resulted in measurable performance degradation. Removing $c(t)$ increased MAE by 33% and RMSE by 37%, indicating loss of day-type discrimination while preserving trend and weekly patterns. Omitting $z(t)$ produced a 102% MAE increase and 95% RMSE increase, demonstrating failure to capture the monotonic trend despite retaining seasonal structure. Eliminating helical coordinates $x(t), y(t)$ resulted in the most severe degradation (282% MAE increase, 246% RMSE increase), producing temporally disordered outputs. Figure 6.3 illustrates these functional deficiencies qualitatively, showing how the complete model reconstructs all temporal structures while ablated variants exhibit specific pattern failures.

Model variant	MAE	RMSE
HTE-cGANs	6.3999	7.9184
w/o $c(t)$	8.5107	10.8748
w/o $z(t)$	12.9533	15.4718
w/o $x(t), y(t)$	24.4730	27.3608

Table 6.6: HTE-cGANs ablation study results. Forecasting accuracy across model variants demonstrates progressive degradation when HTE components are systematically removed. Lower values indicate better performance.

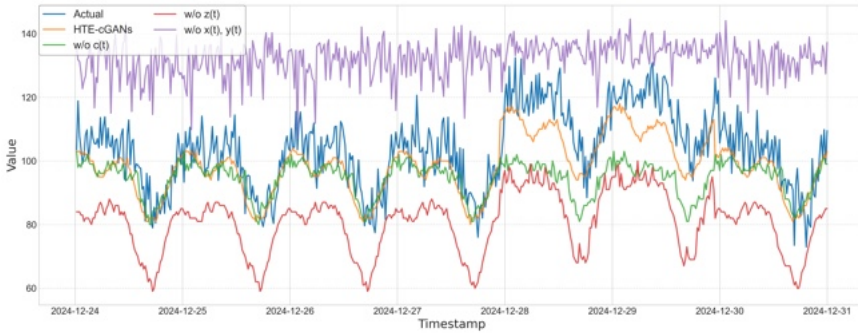


Figure 6.3: Qualitative comparison of the forecasts of the HTE-cGANs model variants on the final week of the test set of the SynthTS dataset.

6.3.2.3 *Robustness Analysis*

We evaluate model stability under two degraded training scenarios: reduced data availability and noisy observations. Table 6.7 presents performance on the TRN-1 configuration (60% of original training data) and TRN-2 configuration (TRN-1 with 2% randomly distributed outliers).

The reduced training set resulted in moderate degradation (7.5% MAE increase, 9.0% RMSE increase), reflecting diminished pattern exposure. The additional noise injection caused limited further degradation (2.7% MAE increase, 2.1% RMSE increase relative to TRN-1),

Training Configuration	MAE	RMSE
Full Training Set	0.1696	0.2322
Reduced Set (TRN-1)	0.1824	0.2531
Noisy Set (TRN-2)	0.1873	0.2585

Table 6.7: Robustness evaluation of HTE-cGANs under reduced training data and noisy conditions on the NYC TLC dataset. Lower values indicate better performance.

suggesting that the adversarial training procedure learns distributional patterns that exhibit some tolerance to individual outliers.

6.3.2.4 *Hyperparameter Sensitivity*

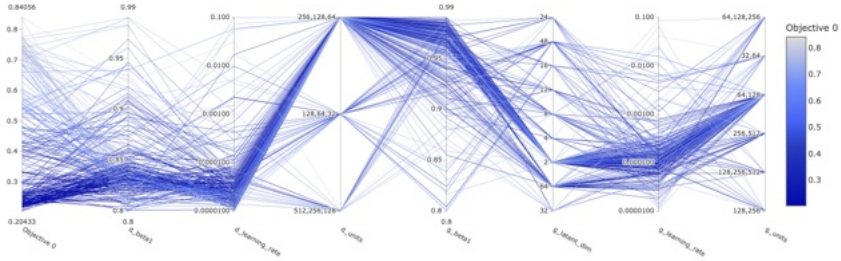
Figure 6.4 shows hyperparameter optimization results across 200 trials for both datasets. The optimization reveals dataset-dependent optimal configurations, with learning rate and the Adam optimizer's β_1 parameter emerging as particularly influential for training stability. Architecture parameters (hidden units, latent dimensions) scale with problem complexity and input dimensionality.

The smoothing factor analysis (Figures 6.5 and 6.6) examines the relationship between sample averaging and prediction stability. Higher smoothing factors aggregate more generated samples per temporal bin, producing more stable predictions as illustrated in Figure 6.7. The optimal factor depends on seasonal period length and required temporal granularity.

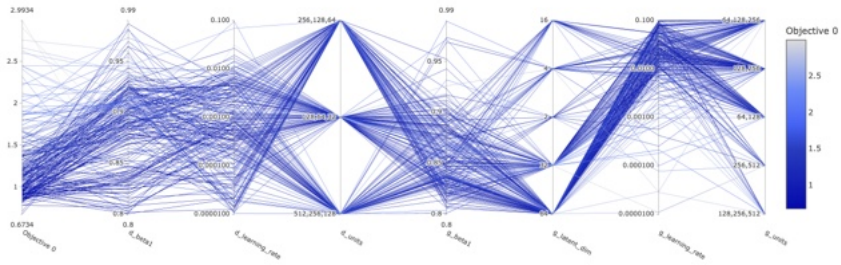
6.3.2.5 *Computational Performance*

Table 6.8 presents training and inference metrics measured on a MacBook Air M1 with 8GB RAM. The MLP-based architecture enables training and deployment on consumer hardware without GPU acceleration.

Figure 6.8 shows the relationship between smoothing factor and inference time. The NYC TLC dataset requires slightly higher compu-



(a) Hyperparameter search for NYC TLC dataset.



(b) Hyperparameter search for ILI dataset.

Figure 6.4: Hyperparameter optimization results for the HTE-cGANs model after 200 trials across both datasets.

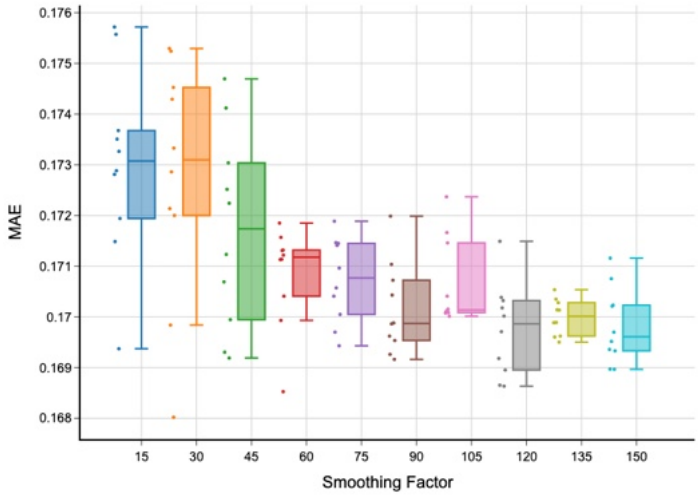
tational overhead due to its temporal frequency (20-minute intervals over 18 days) compared to ILI's weekly resolution.

6.4 DISCUSSION

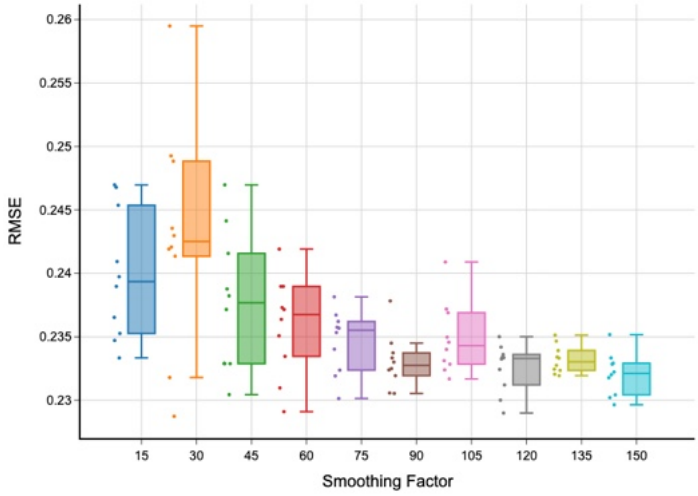
The experimental results provide insights into forecasting with HTE and its implementation through conditional generative models. This section discusses the main findings, contextualizing performance characteristics and discussing practical implications.

6.4.1 *Performance Characteristics and Error Analysis*

The HTE-cGANs model achieves a 70.7% reduction in RMSE compared to OD-TGAT on taxi demand forecasting while maintaining

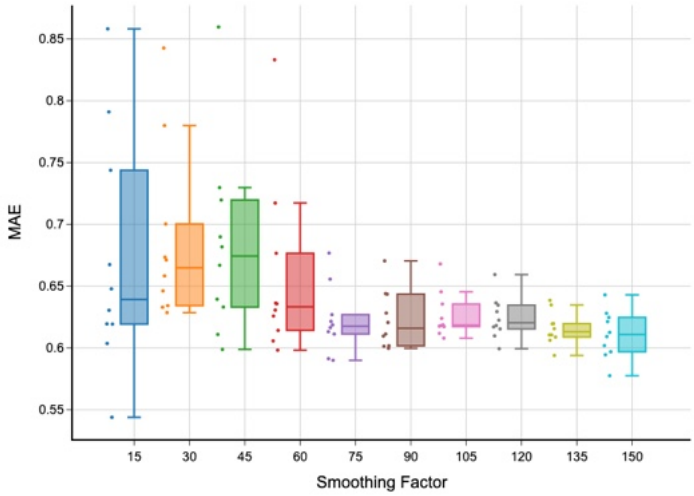


(a) MAE.

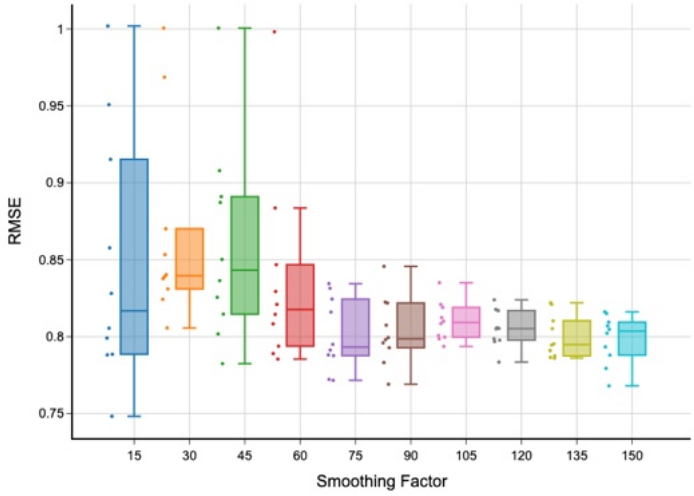


(b) RMSE.

Figure 6.5: Effect of smoothing factor on HTE-cGANs forecasting accuracy for the NYC TLC dataset.

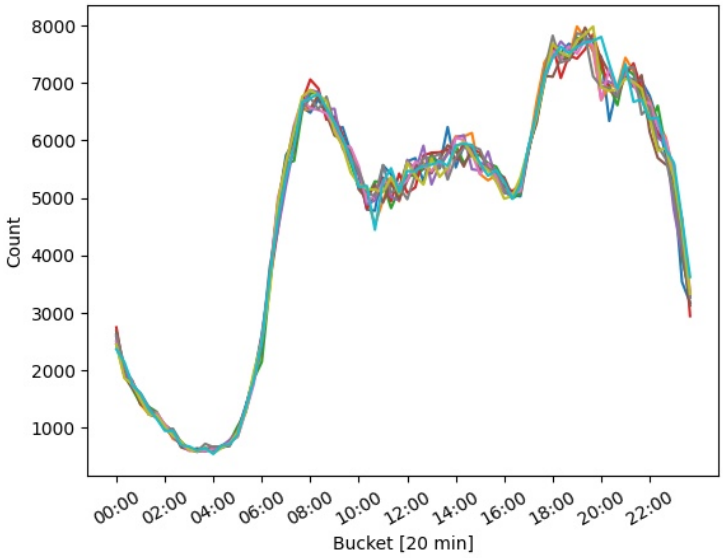


(a) MAE.

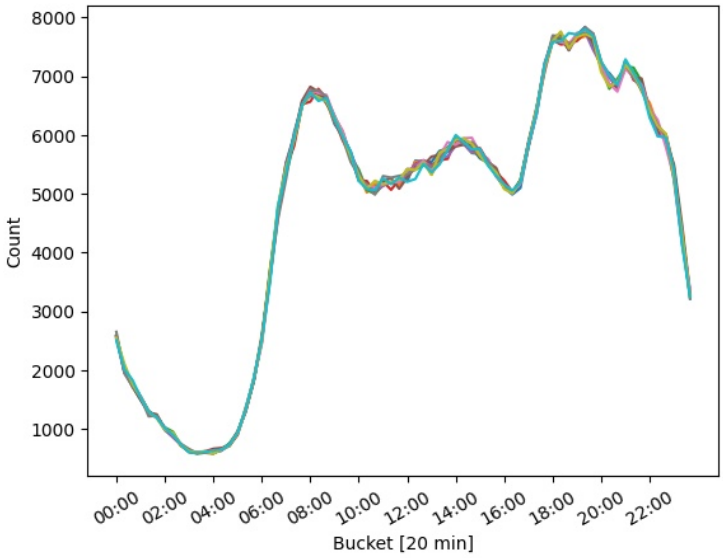


(b) RMSE.

Figure 6.6: Effect of smoothing factor on HTE-cGANs forecasting accuracy for the ILI dataset.

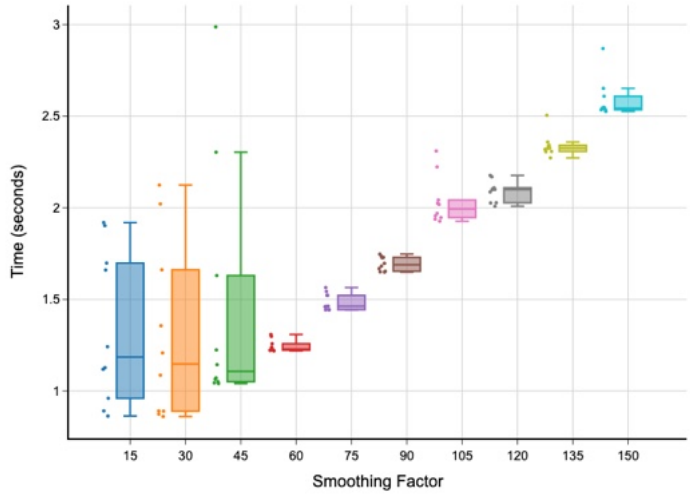


(a) $s = 15$ (1,080 samples).

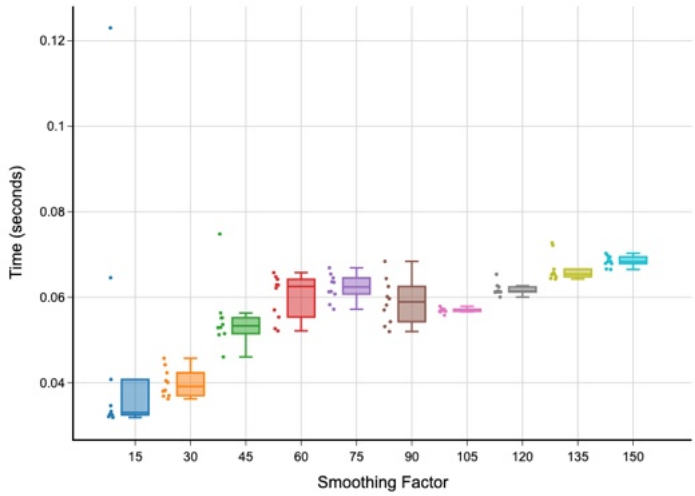


(b) $s = 100$ (7,200 samples).

Figure 6.7: Qualitative comparison of smoothing factor on HTE-cGANs prediction consistency across ten independent generations of the same time series. Higher sampling density produces more consistent predictions across runs.



(a) NYC TLC dataset.



(b) ILI dataset.

Figure 6.8: Impact of smoothing factor on HTE-cGANs inference time.

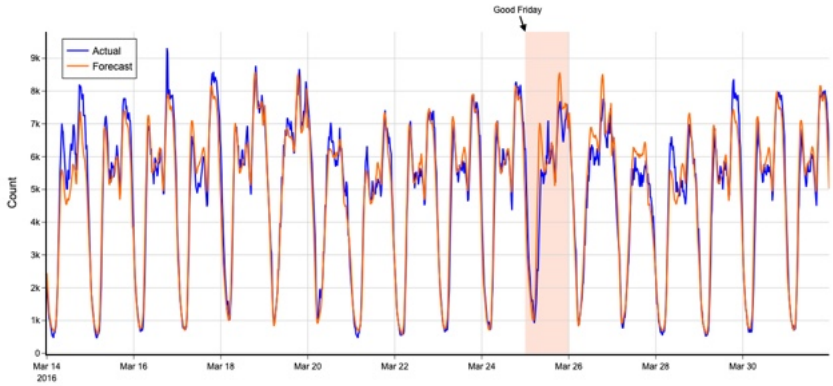
Metric	NYC TLC	ILI
Avg. Seconds/Epoch	0.672	0.483
Num epochs	255	23
Avg. Memory Usage (%)	35.1	38.5
Avg. CPU Usage (%)	60.4	63.2

Table 6.8: Computational performance metrics recorded during training of HTE-cGANs on MacBook Air M1, 8GB RAM.

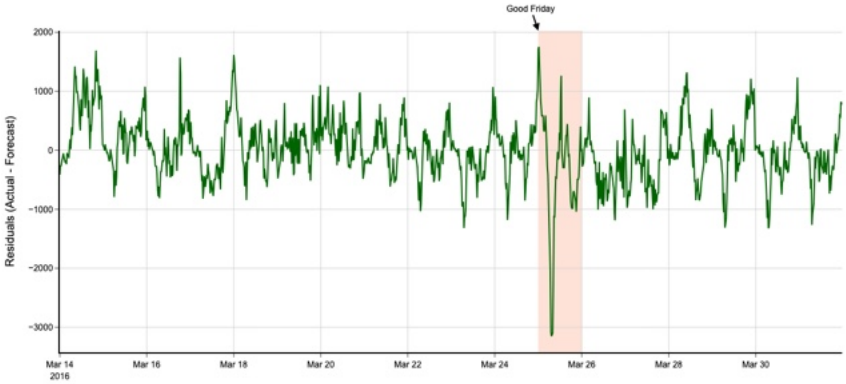
comparable MAE. This asymmetric improvement, characterized by a substantial RMSE reduction accompanied by a modest change in MAE, indicates a decrease in error variance rather than a uniform reduction in error magnitude. Two factors contribute to this stability: first, generative models learn distributional patterns that exhibit inherent tolerance to individual outliers; second, HTE anchors predictions to historical seasonal patterns, enabling the model to recover accurately once values return to learned ranges after transient deviations.

The analysis of forecast residuals (Figure 6.9) reveals that prediction errors are concentrated on Good Friday (25 March 2016), which exhibits atypical demand patterns. Since the training set contains no other instances of this specific holiday, the model treats it as a regular Friday without additional conditioning. Excluding Good Friday from the evaluation yields further improvement, with MAE decreasing from 0.1696 to 0.1599 and RMSE from 0.2322 to 0.2106. Performance is therefore limited primarily by the presence of unrepresented pattern types rather than systematic forecasting deficiencies.

For ILI forecasting under simulated scenarios where future ground truth is unavailable, the model achieves a 42.2% reduction in MAE and a 35.3% reduction in RMSE compared to Autoformer. This advantage stems from the elimination of autoregressive error propagation: whereas sequential models compound prediction errors across time steps, the generative approach produces each forecast independently based on timestamp features. This characteristic renders the ap-



(a) Predicted versus actual taxi demand.



(b) Prediction residuals (actual minus forecasted values).

Figure 6.9: Forecasts of HTE-cGANs on the NYC TLC dataset. Good Friday represents an atypical pattern not captured in $c(t)$, resulting in lower-than-predicted demand.

proach suitable for scenarios requiring extended forecasting horizons or where waiting for incremental observations is impractical.

6.4.2 *Sequential Processing versus Conditional Generation*

Sequential forecasting architectures construct input-output pairs through windowing, a process that requires sufficient historical context for each prediction. The proposed approach eliminates this dependency by encoding individual observations with explicit geometric coordinates, where temporal relationships manifest as spatial relationships within the helical structure. This transformation enables the model to learn from the complete training dataset as independent observations rather than artificially constrained window segments.

This paradigm addresses a limitation of previous GAN-based forecasting attempts, where sliding window preprocessing imposed sequence length constraints that hindered the learning of long-term dependencies. By replacing windowing with geometric encoding, generative models achieve competitive forecasting performance. This suggests that prior limitations stemmed from preprocessing choices rather than an inherent unsuitability of the generative framework for temporal prediction.

The input augmentation experiments with HTE (Table 6.3) demonstrate that HTE also benefits sequential architectures when incorporated as an additional feature, producing consistent improvements across MLP, CNN, and LSTM models. This indicates that explicit temporal representation enhances pattern recognition in standard deep learning architectures without requiring structural modifications, reinforcing the finding that the quality of temporal encoding substantially affects forecasting performance across paradigms.

6.4.3 *Feature Engineering and Domain Transferability*

Models such as OD-TGAT incorporate domain-specific features, including spatial dependencies, Fourier components, weighted moving averages, and neighboring destination data. In contrast, HTE-cGANs achieves competitive performance using only aggregated demand

values and HTE, demonstrating that explicit geometric temporal representation can substitute for elaborate feature engineering in seasonal contexts.

The ablation study (Table 6.6) validates this design choice, demonstrating that each component of the encoding (helical coordinates for order, $z(t)$ for trend, and $c(t)$ for specific conditions) is essential. Removing any single component results in functional failure, confirming that the model relies on the complete geometric representation to resolve trend and seasonality rather than implicitly recovering them from partial data. This simplicity potentially enhances transferability across domains, as the approach requires only the identification of the dominant seasonal period rather than the construction of domain-specific features.

The helical coordinates serve dual purposes: conditioning the generator during training to learn position-value relationships, and facilitating the temporal reconstruction of independently generated predictions through the timestamp trick. This represents a functional shift in how temporal information is used, from sequential context to geometric conditioning, enabling the elimination of autoregressive dependencies during inference.

6.4.4 *Training Stability and Evaluation Methodology*

Unlike computer vision, where consensus exists on evaluation metrics for GANs, the time series domain lacks a shared standard [42]. Loss functions are often architecture-specific and non-generalizable, while the choice of metrics for evaluating quality and diversity of generated temporal data remains fragmented, making objective comparison between different generative models difficult [422]. Moreover, from a training perspective, GANs present three main challenges: non-convergence, vanishing gradients, and mode collapse.

To address these challenges, we first monitored the number of generated points in each temporal bucket. Given the uniform sampling assumption of GenTT, this metric verifies that the model covers the entire time series without collapse. Additionally, we monitored stability using complementary metrics: the Wasserstein distance evaluates

how well generated samples match the target distribution, while MAE quantifies forecast accuracy during training. Together, these signals allow prompt identification of learning failures and enable early stopping.

6.4.5 *Hyperparameter Complexity and Window Optimization*

Sequential forecasting methods require the optimization of sliding window parameters, including window width, stride, and prediction horizon. Tables 6.3 and 6.4 show that these parameters substantially affect prediction quality, with performance variations exceeding 50% across configurations. Insufficient window sizes result in the loss of temporal information, while excessive sizes increase noise sensitivity and computational cost. Identifying optimal configurations typically requires extensive experimentation and domain expertise. These findings demonstrate that an identical architecture configuration yields significantly different performance levels depending solely on the input and output window selection.

The proposed approach mitigates these optimization challenges by requiring only knowledge of the seasonal period, which is typically available a priori in seasonal forecasting applications. This reduction in the hyperparameter space shifts the optimization focus to architecture and training parameters rather than data preprocessing decisions. The comprehensive use of historical data as independent observations enables more robust pattern recognition without the risk of information loss associated with suboptimal window sizing.

6.4.6 *Relationship to Existing Temporal Encoding Methods*

HTE provides deterministic vector representations conceptually similar to Time2Vec but introduces explicit geometric constraints based on known seasonal patterns. While Time2Vec offers flexibility through learnable periodic functions, the deterministic geometric structure of HTE enables specific functional capabilities: one-shot seasonal period generation, order-agnostic prediction with post-hoc reordering via the timestamp trick, and the elimination of sliding window depen-

dencies. This design trades generality for functional advantages in seasonal contexts, positioning the approach as a specialized rather than universally applicable solution.

6.4.7 *Computational Characteristics and Deployment Considerations*

The MLP-based architecture enables training and deployment on consumer hardware without GPU acceleration (Table 6.8), with training times under one minute per epoch on both datasets. The primary computational advantage manifests during inference: independence from recent observations eliminates the need for continuous data ingestion and rolling window updates. This is particularly beneficial in high-frequency forecasting scenarios where repeated autoregressive computations become prohibitive.

The smoothing factor analysis (Figures 6.5 and 6.6) reveals the trade-off between prediction stability and computational cost. Higher smoothing factors aggregate more generated samples per temporal bin, producing more consistent predictions at increased computational expense. The optimal factor depends on the seasonal period length and the required temporal granularity. While the simultaneous generation of large forecast sequences may saturate memory on resource-constrained devices, streaming implementations that yield predictions incrementally can effectively address this limitation.

6.4.8 *Robustness Profile and Pattern Coverage*

Performance degrades moderately under reduced training data (Table 6.7), as diminished observation coverage prevents complete seasonal pattern learning. This degradation is proportional to data loss rather than abrupt, indicating that the model maintains reasonable predictive accuracy even with moderately reduced datasets. The requirement for sufficient pattern coverage is inherent to distributional learning approaches.

Noise injection produces minimal additional degradation beyond that caused by reduced data. This resilience stems from distributional learning: generative models capture underlying patterns rather

than fitting individual observations, providing natural robustness to outliers compared to sequential models that may overfit anomalous observations within training windows.

The response to atypical seasonal patterns reveals operational boundaries. Residual analysis shows that prediction errors concentrate on days exhibiting patterns absent from the training data (e.g., Good Friday). When such atypical instances are excluded from the evaluation, performance improves substantially, indicating that the model performs well when test patterns align with learned distributions. This characteristic is inherent to data-driven approaches and highlights the importance of representative training data.

Unlike sliding window methods that require preprocessing and iterative predictions, the model generates forecasts directly from timestamp-derived conditions. This eliminates warm-up periods and provides inherent robustness to missing recent observations during inference.

6.4.9 *Domain Applicability and Multi-Seasonality*

The approach shows promise for domains involving human activities with clear seasonal cycles, such as retail sales, web traffic, social media activity, energy consumption, and communication patterns. These domains share observable multiple seasonality [176] amenable to geometric encoding such as HTE.

The implementation addresses multiple seasonalities by mapping one dominant pattern to the helix while using additional conditioning mechanisms ($c(t)$, $z(t)$) to capture secondary periodicities and trends. For multivariate signals, the approach assumes shared primary seasonality across variables. Domains lacking clear seasonal structure or exhibiting predominantly irregular patterns fall outside the intended scope of this application.

6.5 THREATS TO VALIDITY

The proposed approach targets time series with observable, consistent seasonality and is designed for scenarios where seasonal pat-

terns remain stable and long-term forecasting is prioritized. The model currently lacks online learning capabilities, requiring retraining when patterns evolve substantially. For applications where recent observations are readily available and short-term forecasting suffices, traditional sequential methods may remain preferable due to their maturity and optimization.

6.5.1 *Internal Validity*

Standard GANs prioritize architectural simplicity but introduce training instability requiring hyperparameter tuning. Figure 6.4 demonstrates that learning rate and Adam's β_1 parameter affect convergence, necessitating careful configuration per dataset. This sensitivity may limit reproducibility without detailed hyperparameter reporting. More stable generative architectures could reduce tuning requirements but were not explored in this work.

The smoothing factor introduces an additional hyperparameter affecting the prediction stability-computational cost trade-off. While analysis reveals systematic relationships between smoothing factor and performance metrics, optimal selection remains dataset-dependent. The experiments employ grid search within computationally feasible ranges, potentially missing optimal configurations outside explored bounds.

6.5.2 *External Validity*

The evaluation focuses on two application domains (urban transportation and public health) with well-defined seasonal patterns. Performance characteristics in domains with irregular seasonality or rapidly evolving patterns remain unexplored. The approach assumes seasonal stability over the forecasting horizon.

The comparison against baseline methods on ILI forecasting simulates scenarios where future ground truth is unavailable by forcing autoregressive prediction. However, baseline methods were originally designed for operational settings where recent observations

are accessible. This comparison demonstrates advantages in specific constrained scenarios.

6.5.3 *Construct Validity*

The approach requires a priori knowledge of the dominant seasonal period. The experiments assume correct seasonal period identification, though misspecification could degrade performance. In domains where this information is unavailable or ambiguous, preliminary seasonality analysis becomes necessary.

HTE explicitly represents one seasonal pattern while relying on the neural architecture to learn additional periodicities implicitly. The relative importance of explicitly encoded versus implicitly learned patterns, and the conditions under which implicit learning suffices, remain incompletely characterized.

6.5.4 *Conclusion Validity*

The model architecture, training procedure, and evaluation protocols introduce multiple sources of variability. While hyperparameter optimization identifies robust configurations, generative model training stochasticity may produce variable results across runs. Additionally, forecasting evaluation metrics such as MAE and RMSE capture point prediction accuracy but do not assess prediction interval coverage, probabilistic calibration, or uncertainty quantification. This choice reflects standard forecasting evaluation practices for direct comparison with existing literature for the considered datasets.

7 TIME SERIES ANOMALY DETECTION

Anomaly detection in time series is challenging in edge environments where rapid, resource-efficient inference is required. Prediction-based methods incur high computational overhead and latency in high-frequency scenarios due to the need to update sliding windows as new observations arrive. Furthermore, no single anomaly detection algorithm is universally effective across domains without adaptation [85, 356]. This chapter addresses RQ4 by instantiating the GenTT framework for anomaly detection on seasonal time series. We introduce RTAD-cVAE, an architecture combining cVAEs with HTE. Unlike standard approaches that continuously generate short predictions, our method directly generates long-term expected patterns, enabling the immediate evaluation of incoming data against pre-computed reference baselines. We evaluate RTAD-cVAE against established baselines in both semi-supervised and unsupervised settings, demonstrating its energy efficiency and robustness to missing data.

7.1 INTRODUCTION

Anomaly detection is essential across finance, cybersecurity, manufacturing, healthcare, smart cities, smart grids, and earth sciences [60, 180, 466]. The proliferation of sensors has generated higher-dimensional data characterized by noise, non-stationarity, and seasonality. These factors, compounded by incomplete or missing observations, complicate the distinction between true anomalies and natural variations.

Classical methods such as k-Nearest Neighbors [126], LOF [48], DBSCAN [111], isolation forests [262], and one-class SVMs [355] detect anomalies through distance or density measurements between data points. While computationally efficient for many applications, these approaches are not designed to capture temporal dependen-

cies and complex patterns inherent in time series data. Additionally, distance-based methods may be hindered by the curse of dimensionality in high-dimensional spaces, where distance metrics become less discriminative.

Real-time anomaly detection commonly employs prediction-based methods that require waiting for new values to generate subsequent forecasts and demand continuous processing of historical windows. However, these models accumulate prediction errors as forecast horizons extend [284], limiting their effectiveness to short-term periods [181, 393, 470]. Reconstruction-based methods leverage access to complete observations to detect deviations, but while this allows for accurate pattern reconstruction, it introduces detection delays that may be unacceptable in real-time contexts [466].

The proposed anomaly detection approach with RTAD-cVAE decouples inference from the conventional sliding window mechanisms of prediction-based methods. By leveraging the geometric properties of HTE, the model generates expected baselines independently of recent history, facilitating zero-latency anomaly detection through direct distance computations. This design ensures real-time operation, robustness to missing data, and computational efficiency suitable for edge deployment. Section 7.2 describes the experimental setup, Section 7.3 presents the results, Section 7.4 discusses performance characteristics and operational considerations, and Section 7.5 examines threats to validity.

7.2 EXPERIMENTAL SETUP

We address anomaly detection in seasonal time series under the following operational constraints: independence from historical data at inference time, robustness to partial or missing streaming observations, computational efficiency for commodity hardware, and offline operation capability. This section presents datasets, baseline methods, evaluation metrics, and implementation details.

Dataset	Origin	Dimensions	Size	Anomalous obs.
SynMul16	Synthetic	16	346k	15%
Dodgers Loop Sensors	Real-world	1	4k	6%
NYC Taxi	Real-world	1	10k	10%
Real Tweets (AMZN)	Real-world	1	16k	10%
Real Tweets (FB)	Real-world	1	16k	10%
Real Tweets (GOOG)	Real-world	1	16k	9%
MIT-BIH Arrhythmia	Real-world	14	167k	1%

Table 7.1: Summary of datasets used for the empirical validation of RTAD-cVAE.

7.2.1 *Datasets*

We evaluated RTAD-cVAE using five univariate seasonal time series from publicly available datasets, one synthetic multivariate dataset designed to test scalability and robustness, and one cardiac dataset to assess applicability boundaries when seasonality is absent (Table 7.1).

SynMul16 is a synthetic dataset comprising 16 channels generated using the GutenTAG tool [424]. It spans 20 days with observations at 5-second intervals, yielding 345,600 timestamps. Anomalies include mean shifts, amplitude and variance changes, platform shifts, extrema events, and frequency perturbations. This dataset is used primarily to evaluate computational performance on MTS.

The Dodgers Loop Sensors dataset captures traffic patterns recorded by loop sensors near the Los Angeles Dodgers stadium. The data cover 25 weeks with measurements at 5-minute intervals, exhibiting strong weekly seasonality influenced by game schedules and attendance patterns. Anomalous behaviors arise from special events, incidents, and unusual crowd dynamics.

The NYC Taxi dataset, from the Numenta Anomaly Benchmark (NAB), contains 10,320 observations of taxi passenger counts at half-hourly intervals. The series displays daily and weekly seasonal components, with anomalies corresponding to unusual events affecting transportation demand, such as major holidays, weather events, and public gatherings.

The Real Tweets dataset, also from NAB, comprises three UTS tracking Twitter mention volumes for Google (GOOG), Facebook (FB), and Amazon (AMZN). Each series contains ≈ 16,000 observations at 5-minute intervals. Anomalies reflect significant corporate events, product launches, news announcements, and viral phenomena triggering abnormal mention rates.

The MIT-BIH Arrhythmia database serves as a stress test for non-seasonal data, providing multivariate cardiac signals. Individual heartbeats were extracted and concatenated across 14 patients to form a composite MTS of 1,163 observations.

7.2.2 *Baseline Methods*

We compared RTAD-cVAE with statistical approaches, classical machine learning, and deep learning architectures.

For SynMul16, we performed an unsupervised evaluation comparing our model with Median Absolute Deviation (MAD-AD) [177], Prophet [397], Calibrated One-class classifier for Unsupervised Time Series Anomaly detection (COUTA) [444], and Deep Isolation Forest (DIF) [443].

For the Dodgers Loop Sensors benchmark, we adopted an unsupervised setting and compared our model with Isolation Forest (iForest) [262], one-class SVM (OCSVM) [355], Piecewise Median Anomaly Detection (Piecewise AD) [411], LSTM-FD [124], LSTM-AD [284], and AD-LTI [434].

For NYC Taxi, we evaluated our model in a semi-supervised setting using both point-based and event-based metrics. We compared our approach with forecasting methods—Moving Average (MA), Seasonal Integrated Moving Average (SIMA), Seasonal ARIMA (SARIMA), Multi-SARIMA [383], and TBATS [93]—as well as anomaly detection approaches such as DeepAnT [297], Skyline [372], and Numenta [4, 218].

For Real Tweets, we compared our method with TadGAN [134] and AER [428] from the Orion benchmark.

For MIT-BIH Arrhythmia, we compared our method in a semi-supervised setting with TimesNet [431], TranAD [408], and Anomaly

Transformer [446], using the implementations provided by the DeepOD [443, 444].

7.2.3 *Evaluation Metrics*

Evaluation metrics for anomaly detection vary significantly across the literature. Some studies use point-based metrics [297], where each detected point is evaluated independently. Others adopt point adjustment (PA) [434], where detecting at least one anomalous point within a window constitutes a true positive (TP). Other studies use event-wise point adjustment (EWPA) [383, 428], which treats entire windows as single anomalies. Several works also employ threshold-free evaluation using AUC-ROC.

To ensure a fair comparison with the literature, we adopted the metrics used in each respective baseline study. We use the subscript e to denote metrics calculated via PA at the event level (i.e., Precision_e, Recall_e, F1_e). We also conducted additional experiments using range-based variants of Precision and Recall ($\beta = 1$, front bias). All metric definitions are provided in Section 2.5.2.

7.2.4 *Implementation Details*

We implemented two RTAD-cVAE configurations based on the architecture in Figure 7.1. For UTS, the encoder uses an MLP with hidden layers of 128 and 64 units, projecting to a 5-dimensional latent space. For the multivariate variant, we increased the capacity using hidden layers of 256, 128, and 64 units, with a 16-dimensional latent space. Both variants use the Adam optimizer with a learning rate of 10^{-3}.

To prevent posterior collapse, we applied KL annealing during the initial 30 epochs, progressively increasing the KL weight from 0 to β. We trained the models for 300 epochs, using a batch size of 32 for UTS and 64 for the multivariate configuration. The reconstruction loss was calculated using MSE. We monitored training stability through the validation loss and used early stopping based on the reconstruction error.

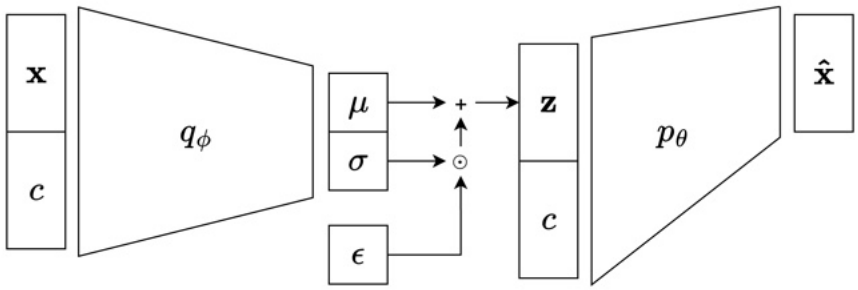


Figure 7.1: The RTAD-cVAE architecture for seasonal time series generation within the GenTT framework.

We calculated anomaly scores using the Manhattan distance and determined thresholds using dataset-specific strategies: peaks-over-threshold [369] for SynMul16; z-score normalization [19] for NYC Taxi and MIT-BIH; and fixed thresholds for Real Tweets (0.7 for AMZN, 0.3 for FB, 0.7 for GOOG).

7.3 RESULTS

In this section, we evaluate the performance of RTAD-cVAE across multiple datasets in both semi-supervised and unsupervised scenarios. We assessed detection accuracy against established baselines, investigated component contributions through ablation studies, and analyzed architectural sensitivity, robustness, scalability, and energy efficiency.

7.3.1 *Anomaly Detection Performance*

We evaluated RTAD-cVAE across datasets with varying seasonal characteristics, comparing it with statistical, machine learning, and deep learning baselines.

Method	TP	FP	FN	Precision_e	Recall_e	F1_e
SARIMA	2	1464	3	0.0014	0.4000	0.0027
SIMA	3	1587	2	0.0019	0.6000	0.0038
TBATS	3	1391	2	0.0022	0.6000	0.0043
Multi-SARIMA	4	1425	1	0.0028	0.8000	0.0056
SIMA+SARIMA	3	1072	2	0.0028	0.6000	0.0056
MA	2	654	3	0.0030	0.4000	0.0061
SIMA+Multi-SARIMA	3	475	2	0.0063	0.6000	0.0124
MA+SARIMA	2	131	3	0.0150	0.4000	0.0290
MA+Multi-SARIMA	2	93	3	0.0211	0.4000	0.0400
NumentaTM	4	178	1	0.0220	0.8000	0.0428
RTAD-cVAE	5	0	0	**1.0000**	**1.0000**	**1.0000**

Table 7.2: Comparison of semi-supervised detection performance on the NYC Taxi dataset using point adjustment (PA) metrics. true positive (TP), false positive (FP), and false negative (FN) represent event counts. Best values in bold.

7.3.1.1 *Semi-supervised Detection*

The NYC Taxi dataset serves as a benchmark with strong daily and weekly seasonality. We performed the evaluation at two levels: event-level detection (identifying anomalous temporal segments) and point-level detection (flagging individual observations).

At the event level, RTAD-cVAE correctly identified all five anomaly events without any false positives. As shown in Table 7.2, statistical models generated hundreds of false alarms while detecting only a subset of true anomalies. The best baseline, Numenta, achieved a precision of 0.0220 and a recall of 0.8000, missing one anomaly event and flagging 178 normal segments as anomalous.

Regarding point-based metrics, RTAD-cVAE attained an F1 score of 0.394 (precision: 0.430, recall: 0.363). Table 7.3 shows that competing methods exhibited extremely low recall. RTAD-cVAE provided a more balanced operating point. Figure 7.2 presents a t-SNE projection of the learned three-dimensional latent space. The visualization reveals a clustering structure corresponding to temporal patterns, demonstrating that RTAD-cVAE learned meaningful representations of the weekly seasonality.

Method	Precision	Recall	F1
Twitter ADVec	0.000	0.000	0.000
Skyline	0.000	0.000	0.000
DeepAnT	**1.000**	0.002	0.004
NumentaTM	0.850	0.006	0.012
Numenta	0.770	0.007	0.014
RTAD-cVAE	0.430	**0.363**	**0.394**

Table 7.3: Comparison of semi-supervised detection performance on the NYC Taxi dataset using point-based metrics. Best values in bold.

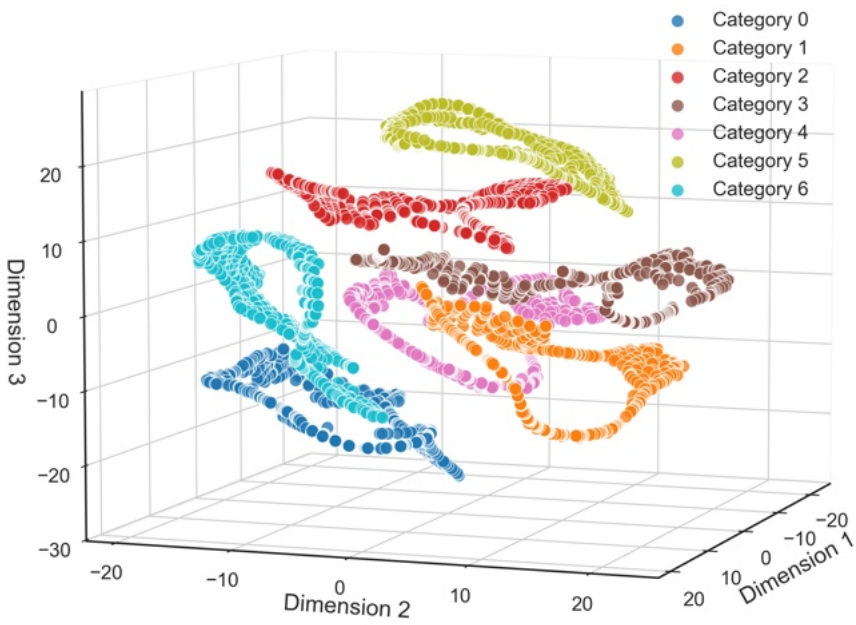


Figure 7.2: t-SNE projection of the latent space learned by RTAD-cVAE on the NYC Taxi dataset. Each category represents a different day of the week.

Method	Precision	Recall	F1
Anomaly Transformer	0.0028	0.5493	0.0056
TranAD	0.0056	0.8028	0.0112
TimesNet	**0.0090**	0.8451	**0.0179**
RTAD-cVAE	0.0033	**0.9156**	0.0066

Table 7.4: Comparison of semi-supervised detection performance on the MIT-BIH Arrhythmia dataset using point-based metrics. Best values in bold.

The MIT-BIH Arrhythmia dataset demonstrates the boundaries of seasonal-based anomaly detection. Due to the absence of seasonal patterns, RTAD-cVAE achieved a precision of 0.0033, while recall remained at 0.9156 (Table 7.4). This result confirms that our approach requires seasonality to constrain predictions. Without such structure, the model defaults to high sensitivity at the cost of specificity.

7.3.1.2 *Unsupervised Detection*

In unsupervised settings, where labeled anomalies are unavailable during training, RTAD-cVAE demonstrated competitive performance on benchmarks with strong seasonal characteristics.

The SynMul16 synthetic multivariate dataset allows for a controlled evaluation with 16 channels. Table 7.5 presents range-based precision and recall metrics. RTAD-cVAE achieved an F1 score of 0.1449, with the highest recall compared to competing methods. Methods such as COUTA and DIF attained a precision exceeding 0.90 but detected fewer anomalous ranges. Prophet achieved comparable behavior to RTAD-cVAE but with lower precision and recall.

The Dodgers Loop Sensors dataset features real-world traffic monitoring with weekly seasonality. Table 7.6 shows that RTAD-cVAE achieved an AUC-ROC of 0.980, outperforming all baselines. Methods such as iForest and OCSVM performed poorly, with an AUC-ROC near 0.5. Figure 7.3 qualitatively illustrates the detec-

Method	Precision_T	Recall_T	F1_T
Prophet	0.3636	0.0765	0.1264
MAD-AD	0.7941	0.0717	0.1315
DIF	0.9091	0.0714	0.1325
COUTA	**0.9365**	0.0714	0.1328
RTAD-cVAE	0.3992	**0.0984**	**0.1449**

Table 7.5: Comparison of unsupervised detection performance on the SynMul16 dataset using range-based metrics. Best values in bold.

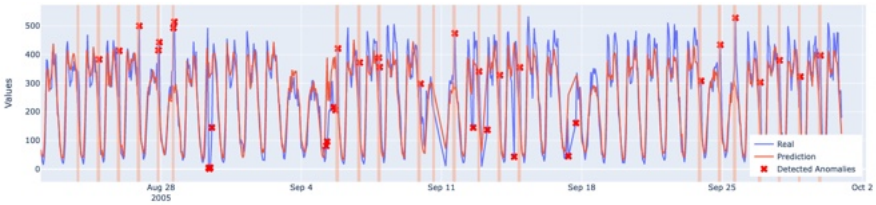


Figure 7.3: Forecast and detected anomalies by RTAD-cVAE on the Dodgers Loop Sensors dataset: actual values (blue), predicted baseline (red), detected anomalies (red markers), ground-truth anomalies (shaded regions).

tion results of RTAD-cVAE, showing a smooth seasonal baseline that adapts to weekly patterns while maintaining sensitivity to deviations.

For the Real Tweets dataset, Table 7.7 presents the event-level metrics. Performance varied across signals: for AMZN, RTAD-cVAE achieved a precision of 1.000 and an F1 score of 0.667, although AER attained higher event recall (i.e., EDR). On the FB signal, RTAD-cVAE outperformed both TadGAN and AER. On GOOG, all methods tied with an F1 of 0.800.

7.3.2 *Ablation Study*

To validate component contributions, we systematically removed temporal conditioning and smoothing to assess their impact on reconstruction accuracy. We performed this evaluation on NYC Taxi,

Method	AUC-ROC
iForest	0.535
OCSVM	0.591
Piecewise AD	0.751
LSTM-FD	0.829
LSTM-AD	0.859
AD-LTI	0.923
RTAD-cVAE	**0.980**

Table 7.6: Comparison of unsupervised detection performance on the Dodgers Loop Sensors dataset using threshold-independent metrics. Best value in bold.

Signal	Method	FP	FN	TP	Recall_e	Precision_e	F1_e
AMZN	TadGAN	0	3	1	0.250	1.000	0.400
	AER	1	1	3	0.750	0.750	**0.750**
	RTAD-cVAE	0	2	2	0.500	1.000	0.667
FB	TadGAN	4	0	2	1.000	0.333	0.500
	AER	3	0	2	1.000	0.400	0.571
	RTAD-cVAE	0	0	2	1.000	1.000	**1.000**
GOOG	TadGAN	0	1	2	0.667	1.000	**0.800**
	AER	0	1	2	0.667	1.000	**0.800**
	RTAD-cVAE	0	1	2	0.667	1.000	**0.800**

Table 7.7: Comparison of unsupervised detection performance on the Real Tweets dataset using point adjustment (PA) metrics. true positive (TP), false positive (FP), and false negative (FN) represent event counts. Best values in bold.

Hyperparameter	Value
Encoder units	256, 128
Decoder units	128, 256
Latent size	16
Learning rate	0.0002
KL annealing epochs	15
KL annealing factor	1.0
Smoothing factor s	20

Table 7.8: Baseline hyperparameters of the RTAD-cVAE model.

limiting training to 30 epochs without early stopping to isolate component effects from optimization dynamics.

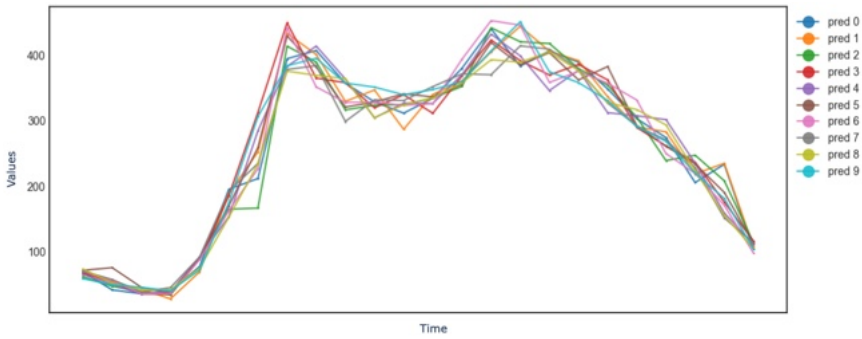
Removing temporal conditioning (categorical information about day of week and time of day) increased the MAE by 0.96% relative to the full model. This difference was statistically significant ($p < 10^{-8}$), confirming a measurable contribution to reconstruction accuracy.

The smoothing factor s, which controls the generated samples per temporal bucket, proved more crucial. Setting $s = 1$ (i.e., removing smoothing) increased the MAE by 59.32% ($p < 10^{-47}$). This demonstrates that timestamp reconstruction and aggregation are essential for producing stable baselines suitable for anomaly detection.

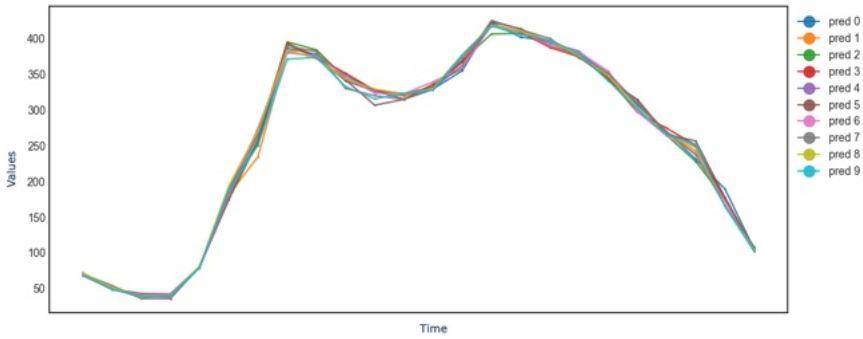
Figure 7.4 illustrates the effect of the smoothing factor. With $s = 10$ (Figure 7.4a), individual generations exhibited substantial variability. With $s = 100$ (Figure 7.4b), aggregation dramatically reduced variability, yielding smooth, consistent baselines that accurately represented the seasonal structure.

7.3.3 *Hyperparameter Sensitivity*

We conducted a sensitivity analysis to explore the impact of model hyperparameters. The baseline configuration (Table 7.8), derived from a tree-structured Parzen estimator, served as the reference point.



(a) $s = 10$.



(b) $s = 100$.

Figure 7.4: Ten seasonal period generations of RTAD-cVAE with different smoothing factors. Higher s reduces variability, stabilizing detection baselines.

We identified distinct configurations for UTS and MTS. UTS benefited from a smaller latent space (2–4 dimensions) and a two-layer architecture (128, 64 units) to prevent overfitting. Conversely, multivariate scenarios necessitated increased representational power to capture cross-channel dependencies. This was achieved using a 16-dimensional latent space and a deeper, three-layer architecture (256, 128, 64 units).

Training dynamics also differed. Multivariate models benefited from lower learning rates ($\approx 2 \times 10^{-4}$) and stronger KL regularization (annealing factors near 1.0) to prevent posterior collapse. Univariate

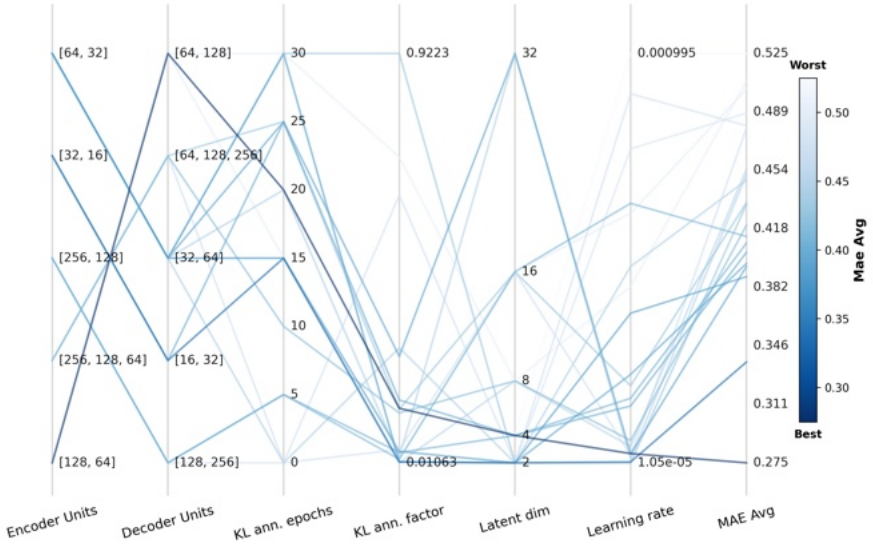


Figure 7.5: Hyperparameter sensitivity of RTAD-cVAE on the NYC Taxi dataset. Darker blue indicates lower MAE.

models preferred weaker KL regularization, focusing on accurate reconstruction without excessive prior constraints.

Figure 7.5 presents a parallel coordinates visualization for NYC Taxi (darker blue indicates lower MAE). The results demonstrate that the lowest reconstruction error corresponds to smaller architectures (128, 64 units) and intermediate learning rates. Figure 7.6 shows the corresponding analysis for SynMul16, where higher dimensionality shifts the optimal configurations toward deeper networks (256, 128, 64 units) and expanded latent spaces.

We also analyzed the sensitivity to training set size on the NYC Taxi dataset (Figure 7.7). Using just 25% of the training data captured most of the achievable performance gains compared to the 10% baseline. Increasing the size to 50% yielded further modest improvements, while using the full dataset provided marginal benefits beyond 50%.

Regarding the smoothing factor s, we observed mild improvements near $s \approx 20$ and $s \approx 140$ on SynMul16, but F1 remained stable from 10 to 160 (Figure 7.8).

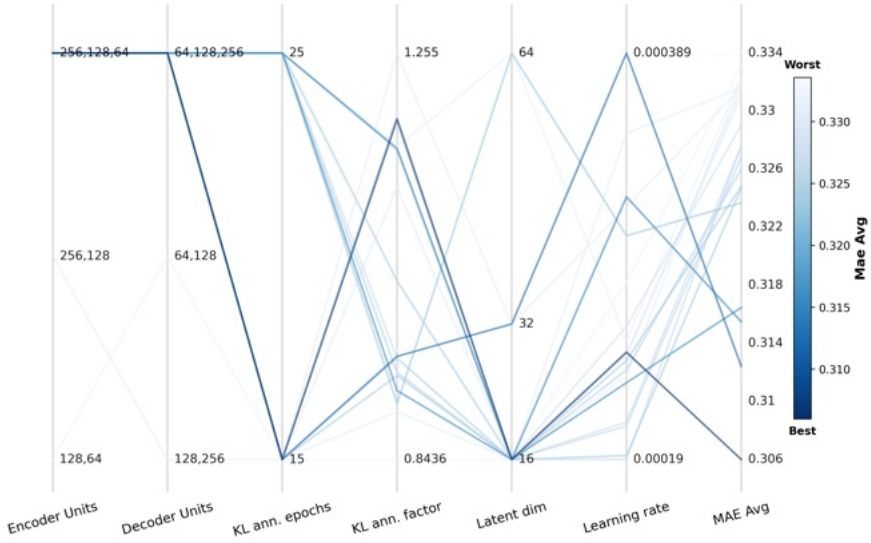


Figure 7.6: Hyperparameter sensitivity of RTAD-cVAE on the SynMul16 dataset. Darker blue indicates lower MAE.

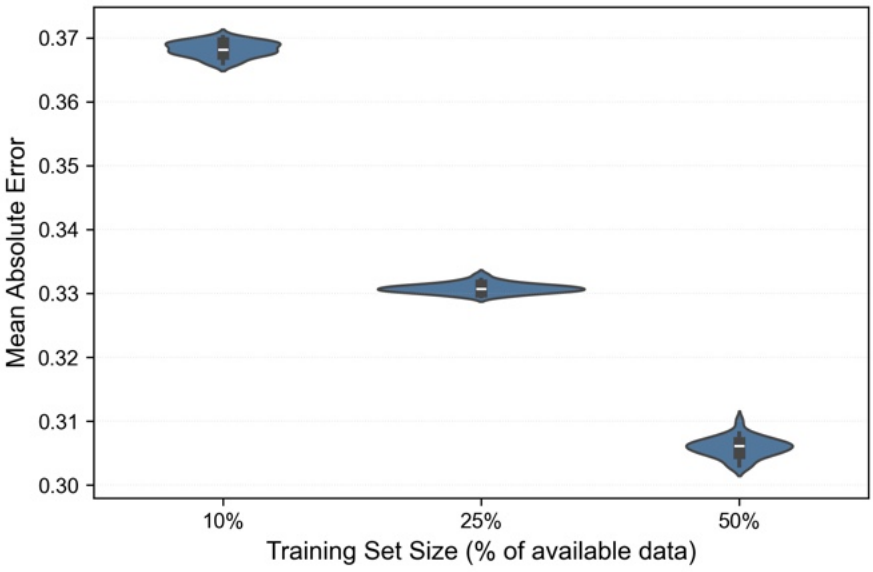


Figure 7.7: Effect of training set size on RTAD-cVAE MAE (NYC Taxi dataset).

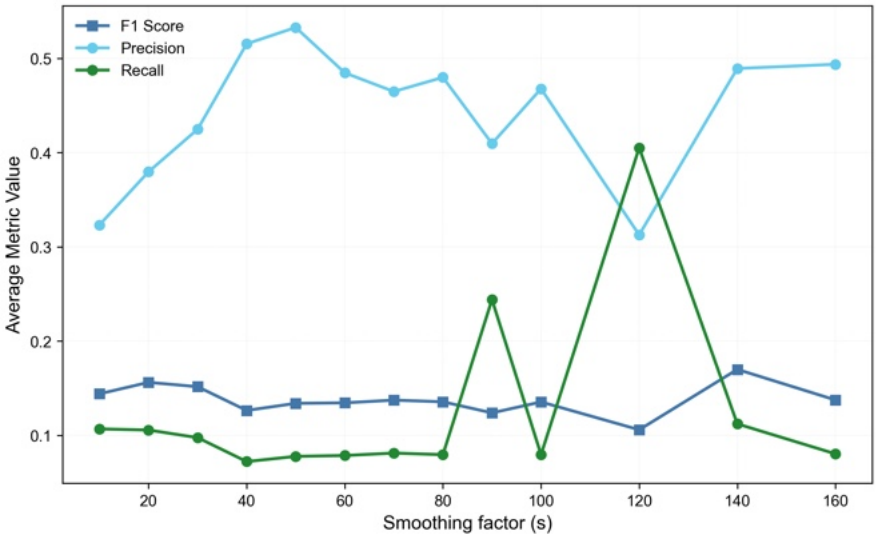


Figure 7.8: Impact of smoothing factor s on RTAD-cVAE anomaly detection metrics (SynMul16 dataset).

Threshold selection analysis compared F1 score distributions across thresholding approaches (Figure 7.9). Decomposition-based thresholding and z-score normalization both performed well with stable, high F1 scores. Peaks-over-threshold showed higher variance, suggesting sensitivity to distributional assumptions, while median absolute deviation consistently underperformed.

7.3.4 *Robustness Analysis*

To assess noise robustness, we injected Gaussian noise at varying intensities. Figure 7.10 displays the F1 distributions at 10%, 30%, and 50% of the feature standard deviation. The model maintained a stable median F1 across all tested levels, demonstrating that seasonal baseline generation remains effective even under substantial noise. However, variance increased at higher noise levels (particularly at 50%), indicating that while the median performance is robust, individual edge cases may degrade.

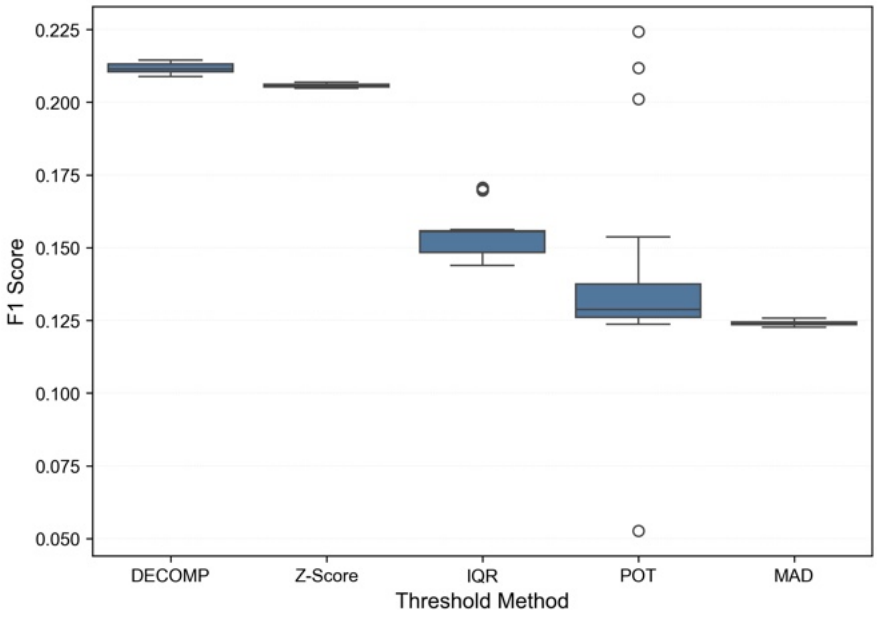


Figure 7.9: F1 score distributions for different thresholding methods used with RTAD-cVAE.

Regarding missing data, we randomly removed observations at rates of 5%, 15%, and 25%. As shown in Figure 7.11, performance remained stable up to 15% missingness. This robustness arises from the conditional generation strategy, which produces seasonal baselines independently of observed data availability. Performance degraded noticeably at 25% missingness, where the reduced number of training examples impacted both learning and threshold calibration.

7.3.5 *Scalability Analysis*

We analyzed how training and inference times scale with system parameters. Training time (Figure 7.12a) grew slowly as the feature count increased. This favorable behavior results from the MLP architecture and the compact latent representation, which does not expand proportionally with input dimensionality. Inference time (Figure 7.12b) remained nearly constant across tested dimensions. This

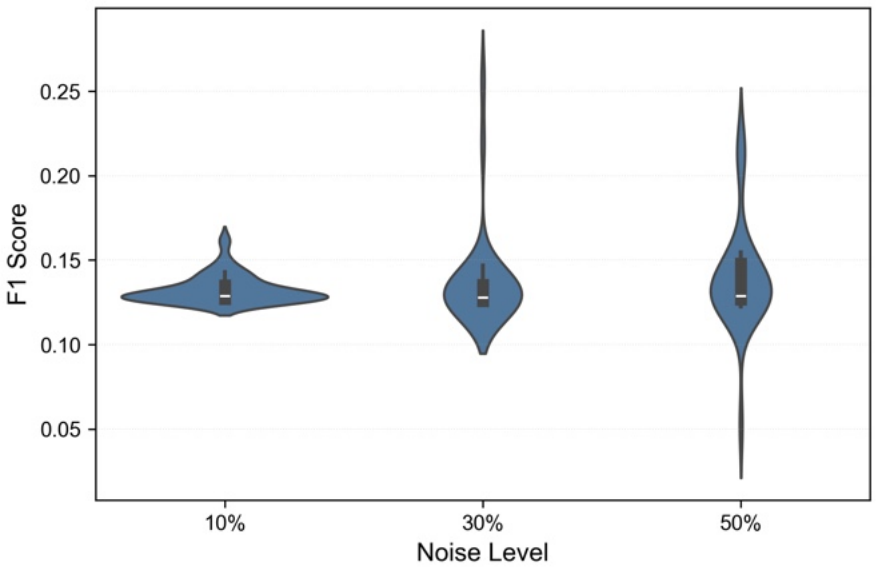


Figure 7.10: RTAD-cVAE noise robustness: F1 distributions at 10%, 30%, and 50% noise levels.

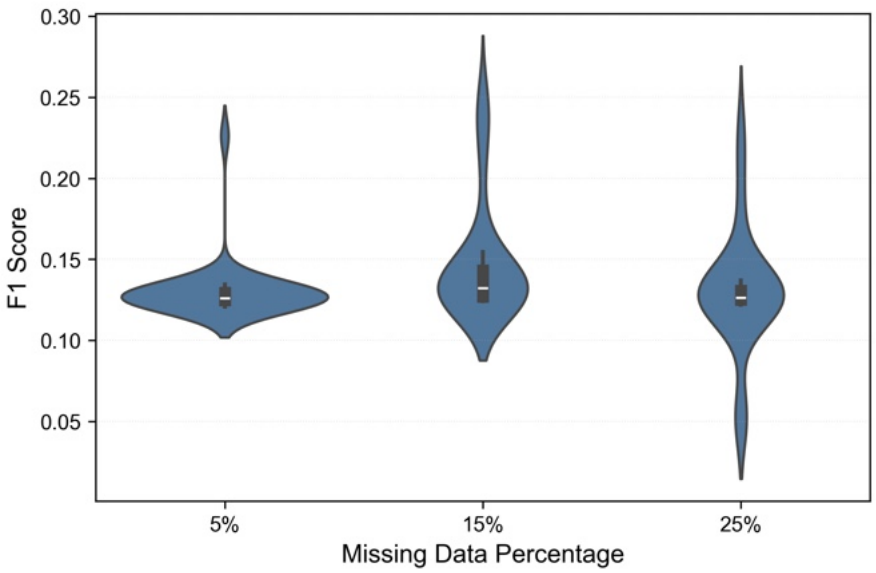


Figure 7.11: RTAD-cVAE missing data robustness: F1 distributions at 5%, 15%, and 25% missingness.

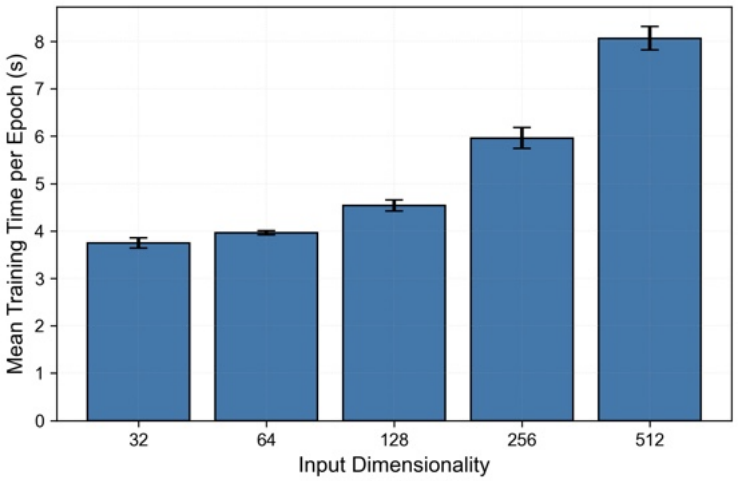
is a significant advantage: once the model is trained, generating seasonal baselines involves sampling from the latent prior and decoding, a process whose complexity depends on the network architecture rather than the input dimensionality.

Conversely, inference time increased with the seasonal period length λ (Figure 7.13). This behavior is expected, as longer periods require generating more temporal buckets. However, absolute times remained modest even for multi-day periods. For extremely long periods, the smoothing factor s can be adjusted to prevent excessive sample generation. Indeed, s acts as the primary computational control, enabling trade-offs between baseline stability and inference speed. Figure 7.14 illustrates its impact: prediction time (Figure 7.14a) increased proportionally with s, while post-processing time (Figure 7.14b) grew rapidly due to sorting and aggregation costs. However, as baseline predictions are generated offline, this cost does not introduce latency during real-time anomaly detection. Given that detection performance remains stable across a wide range of s, values between 20 and 50 offer the best balance between computational efficiency and baseline quality.

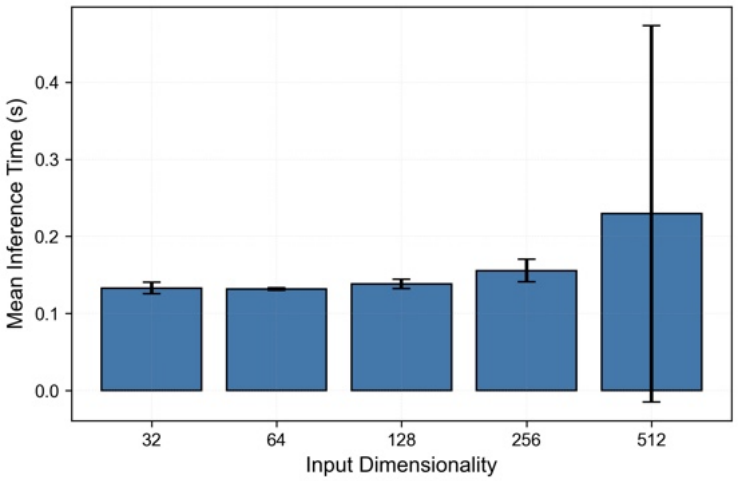
7.3.6 *Energy Consumption and Computational Efficiency*

To assess the energy and environmental footprint, we conducted a streaming simulation on the SynMul16 dataset using a MacBook Air M1 (8GB RAM). We compared RTAD-cVAE with a sliding-window autoregressive long short-term memory (AR-LSTM). Both systems processed data at a 5-second resolution, but their operational logic differed: RTAD-cVAE performed a single long-horizon forecast at the start of each seasonal period, whereas AR-LSTM required full model inference every 5 seconds.

The resulting energy differential was substantial. RTAD-cVAE exhibits a single energy spike during the initial baseline generation, followed by negligible consumption during anomaly scoring. In contrast, AR-LSTM shows continuous power draw. Over a 24-hour period, RTAD-cVAE consumed 8.257×10^{-5} kWh, while AR-LSTM required 0.1037 kWh, representing a 1255-fold reduction (Figure 7.15).



(a) Training time vs dimensionality.



(b) Inference time vs dimensionality.

Figure 7.12: RTAD-cVAE training and inference time scaling with data dimensionality (one-day period).

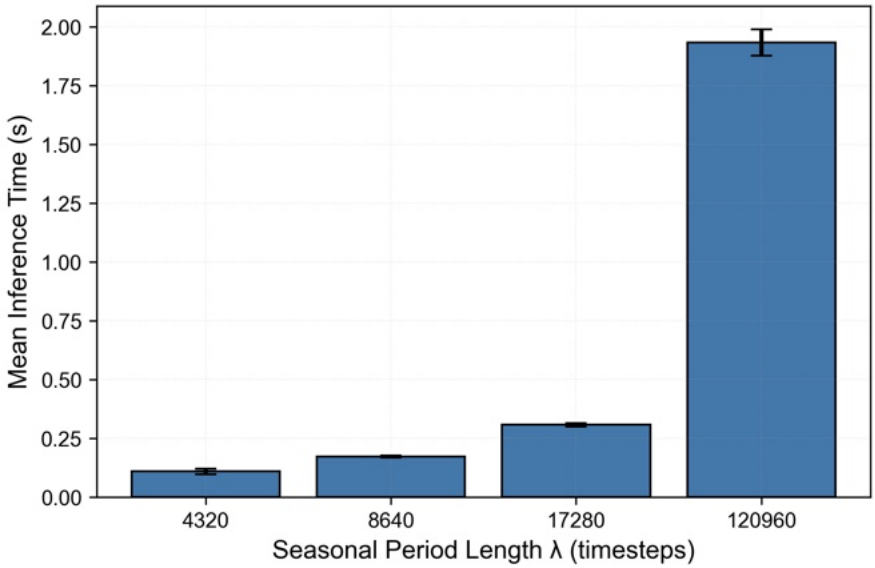


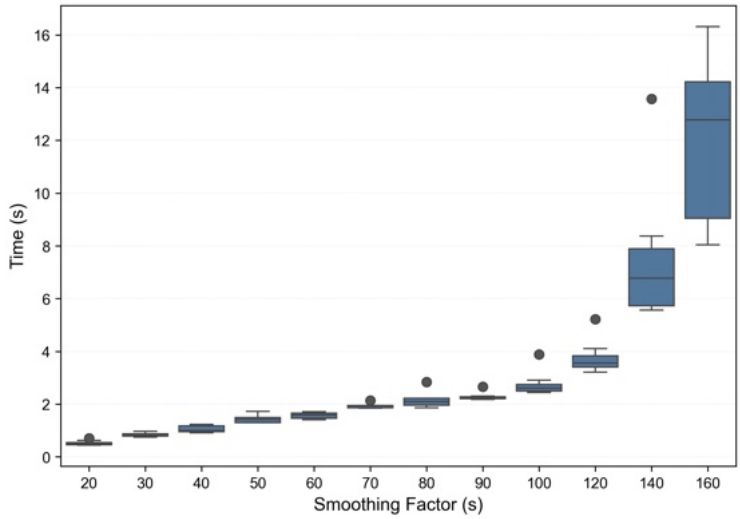
Figure 7.13: Effect of seasonal period length λ on RTAD-cVAE inference time.

This efficiency reduces the carbon footprint: daily CO_2 emissions were 2.731×10^{-5} kg for RTAD-cVAE compared to 0.0343 kg for AR-LSTM (Figure 7.16).

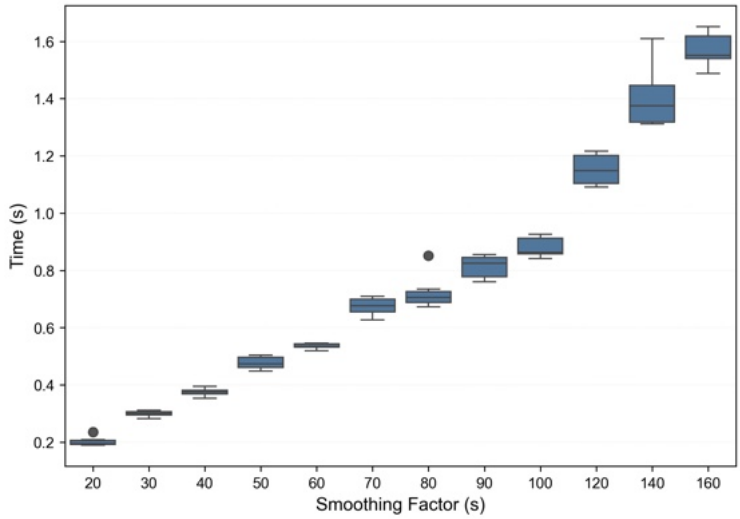
Figure 7.17 illustrates the underlying resource utilization. RTAD-cVAE exhibits a single CPU spike and a constant, low memory footprint, whereas AR-LSTM necessitates frequent CPU spikes and a larger, variable memory buffer to maintain historical data. Such characteristics make RTAD-cVAE well-suited for battery-powered edge devices, resource-constrained IoT deployments, and scenarios where minimizing the carbon footprint is a priority.

7.4 DISCUSSION

The evaluation demonstrates that RTAD-cVAE delivers effective anomaly detection across seasonal time series while ensuring computational efficiency and low energy consumption. The framework achieves real-time processing by generating long-term baseline pre-



(a) Prediction time vs s.



(b) Postprocessing time vs s.

Figure 7.14: Effect of smoothing factor s on RTAD-cVAE processing times (one-day period).

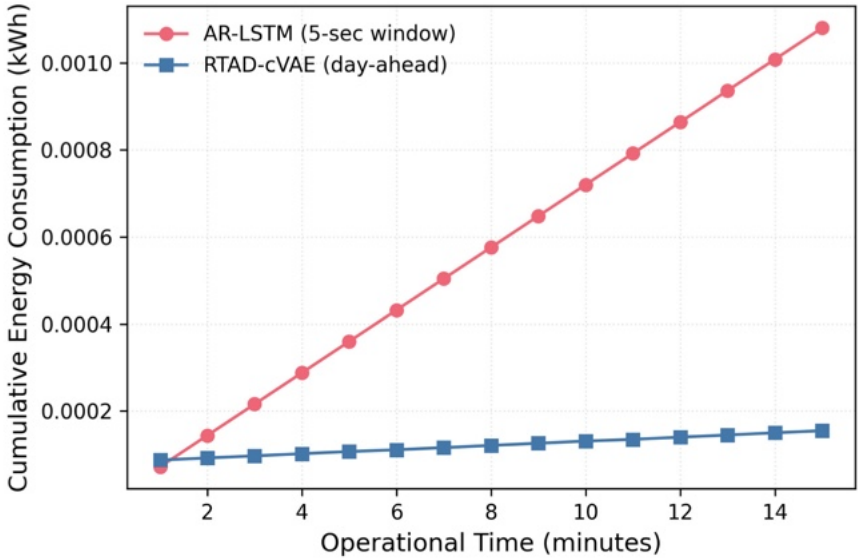


Figure 7.15: Comparison of cumulative energy usage between RTAD-cVAE and AR-LSTM over 15 minutes on SynMul16.

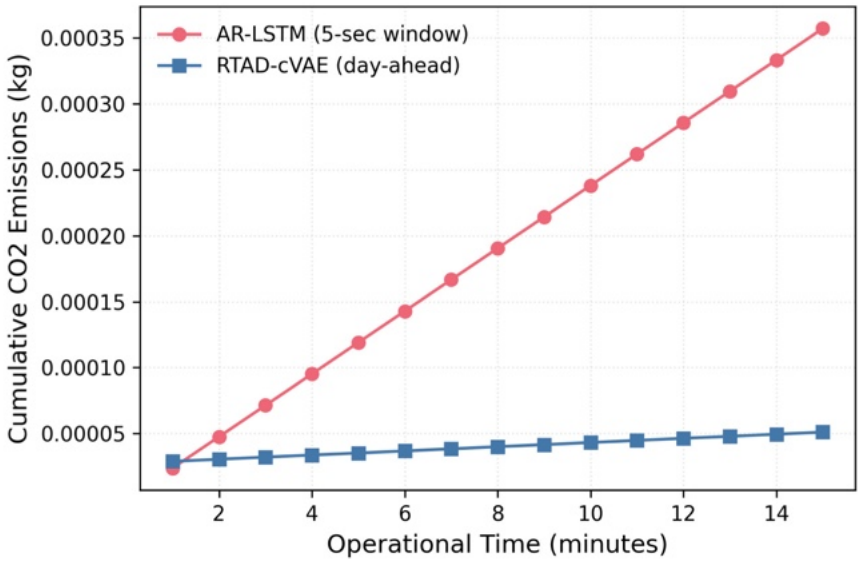


Figure 7.16: Comparison of cumulative CO_2 emissions between RTAD-cVAE and AR-LSTM over 15 minutes on SynMul16.

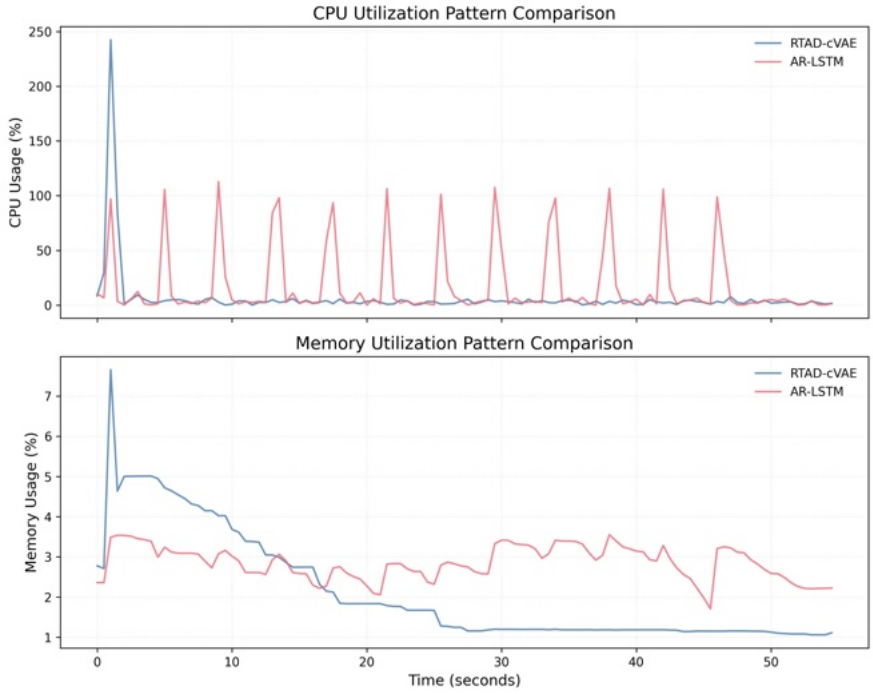


Figure 7.17: CPU and RAM resource usage over one minute on SynMul16. RTAD-cVAE exhibits a single CPU spike, whereas AR-LSTM spikes every 5 seconds.

dictions offline, then performing only lightweight comparison operations as new data arrives.

7.4.1 *Detection Performance and Operational Boundaries*

In semi-supervised scenarios, RTAD-cVAE consistently outperformed baselines on the NYC Taxi dataset, achieving the highest scores in both event-level and point-level evaluations. Similarly, on the Dodgers Loop Sensors dataset, it obtained the highest AUC-ROC among the tested methods, demonstrating its effectiveness for data with strong weekly seasonality. However, the evaluation on the non-seasonal MIT-BIH Arrhythmia dataset validated the operational boundaries of the approach. These results confirm that the model strictly requires a seasonal structure to function correctly, effectively defining the intended application scope of the framework.

In unsupervised settings, the framework maintained competitive performance. On the Real Tweets dataset, the results aligned with established generative methods, showing that RTAD-cVAE captures seasonal patterns as effectively as more complex adversarial or ensemble approaches. On SynMul16, the framework yielded the highest F1-score, indicating a favorable balance between detection sensitivity and false positive control in high-dimensional synthetic scenarios.

Overall, RTAD-cVAE aligns with state-of-the-art anomaly detection performance while providing distinct operational benefits: long-term forecasting capabilities, real-time processing efficiency, and reduced computational overhead. This positions the framework as a viable alternative for seasonal time series, particularly in resource-constrained environments.

A limitation of RTAD-cVAE is that it does not automatically adapt to abrupt concept drift [122]. This issue can be mitigated through MLOps practices, such as frequent retraining and drift detection metrics to assess data divergence [53, 406]. The lightweight architecture facilitates rapid retraining cycles. Chapter 8 discusses further optimization using MoE architectures for partial model retraining strategies.

7.4.2 *Computational Efficiency and Environmental Impact*

Quantitative measurements showed a 1255-fold reduction in energy consumption compared to AR-LSTM sliding window approaches: RTAD-cVAE consumed 8.257×10^{-5} kWh versus 0.1037 kWh required by AR-LSTM during a 24-hour operation period. This efficiency stems from the long-term prediction approach based on GenTT, which generates robust long-horizon forecasts. By concentrating the computational effort during the initial baseline generation rather than performing short-term sliding window predictions, RTAD-cVAE limits CPU activity to a single initial spike, followed by minimal load during anomaly scoring.

The environmental impact assessment revealed corresponding benefits: RTAD-cVAE produced 2.731×10^{-5} kg of CO_2 emissions compared to 0.0343 kg for the sliding window approach. When projected across large-scale monitoring systems or extended timeframes, these reductions translate into substantial environmental benefits, making RTAD-cVAE particularly suitable for long-term monitoring where energy consumption represents a significant cost.

7.4.3 *Robustness and Data Quality Tolerance*

RTAD-cVAE demonstrated resilience to common data quality issues, maintaining stable performance under increasing noise and missing data. Our results indicate that the learned latent representations successfully capture seasonal patterns, which remain identifiable despite local perturbations. The model maintained stable performance with up to 15% randomly distributed missing observations, with reliability degrading only when missingness exceeded 25%.

This tolerance arises from the generative model's ability to learn underlying seasonal patterns from the overall distribution of the training data rather than relying on precise point-by-point correspondences. Consequently, the latent space representations are less susceptible to local missing values, ensuring reliable detection even in environments where perfect data quality cannot be guaranteed.

7.4.4 *Scalability Profile*

The scalability analysis revealed favorable computational characteristics. Training time grew slowly relative to dimensionality, suggesting that the RTAD-cVAE architecture effectively leverages shared representations across dimensions. This avoids the computational explosion typically associated with processing high-dimensional time series.

Inference time remained largely constant regardless of input dimensionality. This constitutes a significant advantage over methods where prediction complexity grows with the feature count. The dimension-independence stems from the generative approach, where complexity depends primarily on the temporal bucket count.

However, we identified the smoothing factor as the primary computational bottleneck for nodes with limited RAM. In high-dimensional scenarios, large smoothing factors can create memory pressure. While this can be addressed through generation process optimizations, it remains an important consideration for deployment in resource-constrained environments.

7.4.5 *Architectural Components and Configuration*

The ablation study confirmed the contribution of the proposed components. While explicit temporal conditioning improved reconstruction accuracy, the smoothing mechanism proved to be the most critical element. Removing the smoothing factor (setting $s = 1$) resulted in a 59.32% increase in MAE, demonstrating that aggregating stochastic samples is necessary to stabilize the generated baselines.

The hyperparameter sensitivity analysis showed that univariate seasonal data benefit from simpler configurations with lower regularization, which allow the model to capture patterns without overfitting. Conversely, multivariate time series require higher-capacity models with stronger regularization to capture inter-dimensional dependencies. Additionally, the analysis of thresholding methods indicated that decomposition-based and z-score methods offer greater stability compared to peaks-over-threshold.

7.5 THREATS TO VALIDITY

The proposed anomaly detection approach targets time series with observable, consistent seasonality and is designed for scenarios where seasonal patterns remain relatively stable. To prioritize real-time processing and minimal computational overhead, the model operates without online learning capabilities, instead requiring retraining when patterns evolve substantially.

The computational advantages during inference, particularly the independence from recent observations, are most relevant in high-frequency monitoring scenarios, battery-powered edge devices, or when data acquisition delays are prohibitive. In contexts where temporal or computational constraints are absent, traditional sequential methods may remain preferable due to their maturity and adaptive capabilities.

7.5.1 *Internal Validity*

We evaluated multiple thresholding strategies and reported the best performance for each dataset. This selection process introduces an optimistic bias, as the optimal method was identified ex post. In operational deployments, practitioners must select a thresholding mechanism a priori, potentially choosing one that underperforms on their specific data distribution. Consequently, real-world detection performance may be lower than the experimental findings if the selected thresholding strategy is suboptimal.

The field of anomaly detection lacks consensus on evaluation protocols, label definitions, and threshold calculation methods. Existing benchmark datasets have been criticized as trivial, mislabeled, or unrealistic [433], with manual labeling susceptible to subjective judgments due to the absence of unified formal definitions of anomalies. While we acknowledge these concerns regarding benchmark quality, we employed these commonly used datasets to enable direct comparison with competing methods in the literature. Part iv evaluates the approach on real-world payment system data, providing validation beyond academic benchmarks in an industrial setting.

7.5.2 *External Validity*

The evaluation focuses on datasets with identifiable seasonal patterns across multiple domains, such as transportation, social media, traffic monitoring, and synthetic benchmarks. The MIT-BIH Arrhythmia results demonstrate expected performance degradation on non-seasonal data, as the framework is intentionally designed to exploit the structure of seasonal time series.

7.5.3 *Construct Validity*

The experiments assume correct seasonal period identification; misspecification of this parameter degrades performance. Results demonstrate effectiveness when primary seasonality is known and strong. For multivariate time series, the current implementation assumes signals share at least one common seasonal pattern to enable timestamp reconstruction. When common seasonality is absent, individual models must be trained for each signal.

7.5.4 *Conclusion Validity*

Model architecture, training procedure, and evaluation protocols introduce multiple sources of variability. While hyperparameter optimization identified robust configurations, the stochastic nature of generative model training may produce variable results across runs. The energy consumption analysis compared RTAD-cVAE against AR-LSTM under specific conditions: 5-second resolution, 24-hour period, MacBook Air M1 hardware. While the 1255-fold reduction is substantial, precise efficiency gains may vary with different resolutions, seasonal periods, or hardware platforms.

Part IV

APPLIED RESEARCH

Theory must survive the real world.
Where milliseconds determine success.

8 STREAMING ANOMALY DETECTION IN INSTANT PAYMENTS

Prediction-based anomaly detection methods are well-suited for real-time applications as they provide immediate detection capabilities [172, 180, 434]. However, in high-frequency time series such as instant payment volumes, forecast accuracy degrades as the prediction horizon extends due to error accumulation [181, 393, 470], while sliding window models tend to adapt to and mask emerging anomalies. Instant payment data exhibit high-frequency noise and unpredictable spikes that challenge standard predictive models. Addressing RQ5, this chapter extends the generative framework proposed in Chapter 7 into a MoE architecture tailored for this domain. We propose a design where multiple conditional generative models serve as specialized experts. This approach mitigates error accumulation and prevents anomaly adaptation. Furthermore, it enables partial model retraining by updating specific experts independently, maintaining the computational efficiency required for high-frequency streaming anomaly detection in resource-constrained edge environments.

8.1 INTRODUCTION

In real-time systems, correctness depends on both result accuracy and delivery timeliness; delayed responses may render outputs ineffective or hazardous [54]. The financial sector increasingly operates under such constraints. Instant payment systems, for instance, eliminate settlement delays and enable continuous, near-instantaneous fund transfers [103].

CSMs constitute the key infrastructure for instant payments, enabling settlement between payment service providers (PSPs) of end users. The SCT Inst represents a harmonization effort for instant

payment systems across Europe [117]. The primary CSM infrastructures compliant with the SCT Inst scheme [87] are RT1 [80, 102] and TIPS [113, 335]. These systems process millions of transactions daily, settling each within milliseconds.

Processing such high-frequency volumes requires immediate responsiveness to maintain operational integrity. Traditional batch processing introduces latency, delaying identification and negatively affecting metrics such as MTTD and mean time to respond (MTTR). This challenge is particularly acute in resource-constrained environments where accuracy competes with efficiency. While lightweight edge processing offers advantages in latency reduction, energy efficiency, and data privacy [358], storage limitations impose strict trade-offs. For instance, sliding windows may be too short to capture long-term patterns, whereas incremental processing often lacks sufficient seasonal context.

Existing methods face distinct limitations in this context. Statistical methods often struggle with nonlinear high-frequency patterns, while deep learning models frequently demand computational resources exceeding operational constraints for continuous real-time inference. Furthermore, sliding window-based approaches risk absorbing anomalies into their history, effectively masking abnormal behavior.

We propose a MoE architecture based on GenTT with HTE for streaming anomaly detection in high-frequency time series within resource-constrained operational environments. The architecture addresses key deployment challenges: operation in virtualized environments without GPU acceleration, robustness to missing or sparse training data, functioning during data unavailability periods, and reducing training time and energy consumption through selective expert updates rather than complete retraining when model drift occurs.

The remainder of this chapter is organized as follows. Section 8.2 contextualizes instant payment systems within the broader payment infrastructure. Section 8.3 presents the proposed MoE architecture. Section 8.4 describes the experimental setup, benchmarking VAE and GAN instantiations against prediction-based methods using real-world TIPS payment volumes with publicly declared anomalies.

Section 8.5 reports the results, and Section 8.6 discusses findings and limitations.

8.2 INSTANT PAYMENT SYSTEMS

Payment systems are fundamental infrastructures for modern economies, enabling banks, financial institutions, public administrations, businesses, and citizens to conduct transactions involving fund transfers, thereby settling obligations between payers and beneficiaries [489].

Payment systems can be classified by their settlement method into net and gross systems [303]. Net settlement systems base settlement on net positions calculated as the sum of amounts received minus amounts paid. Gross settlement systems settle the gross amount of each individual payment order. Traditionally, gross settlement systems handle large-value fund transfers between banks in real time, hence the term real-time gross settlement (RTGS) systems [83, 84]. An RTGS system is typically designed as a large centralized system prioritizing service quality in terms of latency and reliability.

However, traditional RTGS architecture is not well-suited for large volumes of retail payments [11]. The retail payment market still relies on batch-based clearing systems, which group multiple payments and send a single payment request to the RTGS, allowing thousands of retail payments to complete with one transaction executed by the RTGS.

While netting infrastructures handle high transaction volumes, their latency can extend to hours or days, and money exchange is limited to commercial bank working hours. Consumers increasingly expect convenient and faster services, such as making internet purchases anytime and anywhere, including evenings, weekends, and holidays when most traditional electronic payment systems are non-operational. Suppliers seek assurance of immediate payment upon selling goods and services.

An instant payment[1] addresses both needs as a transaction involving multiple participants that completes within seconds and can

1 Also known as real-time, immediate, rapid, or retail fast payment.

occur from any location, 24 hours a day, 7 days a week, 365 days a year.

As of May 2024, instant payments are available in 80 countries [413], with global transactions reaching 195 billion in 2022 (a 63.2% increase from 2021) and projected to hit 511.7 billion by 2027 [1]. India dominates with 46% of global volume (89.5 billion transactions, 76.8% year-over-year growth), followed by Brazil (29.2 billion, 228.9% year-over-year growth), China (17.6 billion), Thailand (16.4 billion), and South Korea (8 billion).

In Europe, instant payment transactions are expected to grow from 13.2 billion in 2022 to 34.2 billion by 2027, driven by strong adoption in the Netherlands, Sweden, Finland, and Denmark [1, 115]. The EU Instant Payments Regulation [116] mandates availability across all 27 member states at costs equal to or lower than standard transfers, further accelerating adoption.

Although no truly global instant payment systems exist, numerous domestic systems operate successfully within individual countries or regions, facilitating cross-border transactions. The key infrastructure is the CSM, which enables settlement between PSPs of end users. In Europe, two main CSMs, RT1 [80] and TIPS [113, 335], comply with the SCT Inst scheme [87].

This dissertation focuses on TIPS, which operates on reactive application principles and leverages a distributed, event-driven architecture capable of processing an average of 500 payments per second, with peak capacities reaching 2,000 payments per second [11, 56]. While instant payment systems typically operate within specific jurisdictions, TIPS represents an exception in Europe by supporting multiple currencies.

Since February 2024, the Sveriges Riksbank RIX-RTGS real-time gross settlement system has been fully integrated into TIPS [112], extending its reach beyond the euro area to include Swedish krona. This connection enables payments through Swish (the Swedish mobile payment solution) using the single instructing party settlement model [114].

The following section introduces a MoE model with generative models based on GenTT and HTE for detecting anomalies in volumes of settled instant payments. The case study involves instant payments

in Swedish krona, for which publicly reported anomalies from Swish are available [135]. The proposed solution enables real-time anomaly detection with low resource requirements and is optimized for deployment after model drift.

8.3 MIXTURE OF GENERATIVE EXPERTS WITH HELICAL TIME ENCODING

We propose a hard-gated MoE architecture where each expert is a conditional generative model specialized on distinct temporal patterns, such as specific days of the week. A hard routing mechanism determines which expert activates for each input, with conditions managing exceptions such as holidays or paydays.

This design leverages HTE properties to capture both seasonality and trends while ensuring continuity between patterns predicted by different experts, enabling effective long-term seasonal time series forecasting with smooth transitions. A key advantage is targeted updates when model drift affects specific patterns, requiring retraining only the relevant expert rather than the entire model.

8.3.1 *Problem Definition*

We address anomaly detection in high-frequency data streams from instant payment systems, specifically identifying unusual drops in transaction volumes. This requires real-time processing of arriving data points with minimal computational resources and low detection latency.

A data stream consists of an unbounded sequence of observations arriving sequentially over time, represented as

$$\{x_1, x_2, \ldots, x_t, \ldots\},$$

where x_t denotes the observation at time step t. Unlike fixed-length datasets, streams have no predefined endpoint and continuously produce new data points.

Anomaly detection in data streams involves real-time identification of observations or subsequences exhibiting substantial deviations

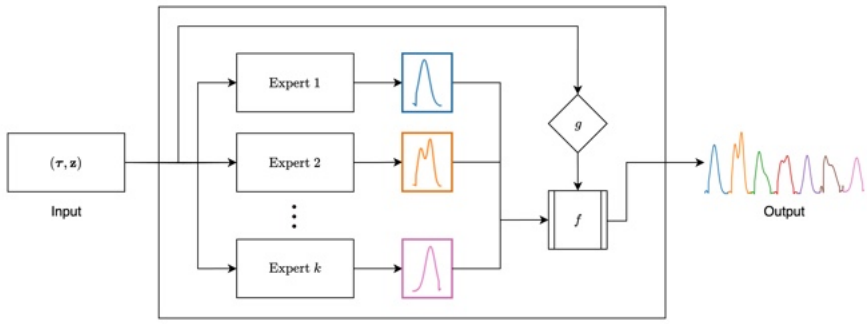


Figure 8.1: Architecture of the hard-gated MoGE with HTE. The model receives timestamp vector $\boldsymbol{\tau}$ and latent vector $\mathbf{z}$ as inputs and generates time series predictions as output.

from expected patterns as data arrives. We approach this as an iterative forecasting problem with prediction-based models, continuously forecasting expected future values and comparing them with actual observations, flagging significant deviations as potential anomalies.

8.3.2 *Mixture of Generative Experts Architecture*

Originally proposed by Jacobs et al. [178], the mixture of experts (MoE) architecture distributes computational tasks among multiple specialized subnetworks, employing a gating network to selectively activate appropriate experts for each input, whose predictions are then combined to form the final output. In scenarios where only a subset of experts is active for any given prediction, this strategy is known as specialized or sleeping experts [40, 127].

Our proposed architecture, mixture of generative experts (MoGE), integrates sleeping expert activation with the GenTT framework. The prediction model comprises k generative experts, a rule-based gating function $g(\cdot)$ that selects the active expert for each prediction, and a combination mechanism $f(\cdot)$ that leverages HTE properties to generate the final forecast (Figure 8.1).

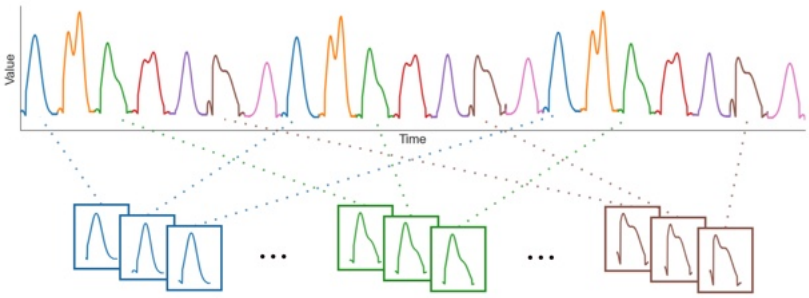


Figure 8.2: Partitioning of a univariate time series based on function $\pi(\tau)$. Each color represents a specific pattern type.

8.3.2.1 *Generative Experts*

Each i-th expert model is a conditional generative model following the GenTT framework using HTE. The expert takes as input a latent vector $\boldsymbol{z}$ and the condition $c(\tau)$ on timestamp τ defined by HTE:

$$f_i(\boldsymbol{z}, c(\tau)) \tag{8.1}$$

The properties of HTE enable capturing global temporal progression even though individual expert models observe only a fragment of the time series. This enables consistent final forecasts capturing both seasonal and trend components. Each expert i is trained independently on the data subset:

$$\mathcal{D}_i = \{(\tau_j, y_j) \in \mathcal{D} : \pi(\tau_j) = i\}, \tag{8.2}$$

where $\pi(\tau_j) : \mathbb{Z} \to \{1, 2, ..., k\}$. For instance, Figure 8.2 shows partitioning of a time series for 7 experts, where function $\pi(\tau_j)$ returns the day of the week, assuming the week starts on Monday (denoted by 1) and ends on Sunday (denoted by 7).

8.3.2.2 *Gating Function*

The gating function $g(\cdot)$ determines which expert is most suitable for a given input timestamp. It is a deterministic function defined for each timestamp $\tau_i \in \boldsymbol{\tau}$:

$$g(\tau_i) = \pi(\tau_i) \tag{8.3}$$

For each input timestamp, only one model is used while others remain inactive. The forecast for input timestamps $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_n)$ and associated latent vectors $\mathbf{z} = (z_1, \ldots, z_n)$ is the temporal concatenation of individual expert predictions:

$$f(\boldsymbol{\tau}, \mathbf{z}) = \bigoplus_{i=1}^{n} f_{g(\tau_i)}(z_i, c(\tau_i)) \tag{8.4}$$

where $\bigoplus$ denotes concatenation of predictions.

8.4 EXPERIMENTAL SETUP

We detect anomalies in high-frequency data streams by identifying drops in settled instant payment volumes from a CSM perspective. When transaction counts in a time bucket are unusually low, this may indicate issues with a SCT Inst participant or problems in the CSM itself. We used a real time series from TIPS [11, 56, 335] involving payments in Swedish krona, for which Swish maintains a public registry of incidents where transactions are impacted [135].

Experiments were conducted in constrained environments where computational and storage resources are limited, and each incoming observation must be evaluated in near real-time. The approach employs prediction-based anomaly detection with iterative long-term forecasting, which provides interpretable results and allows instantaneous evaluation of incoming observations.

Two instantiations of the proposed MoGE architecture, MoGE-cGANs and MoGE-cVAE, are compared against baseline methods that perform predictions in both autoregressive and simulated real-time forecasting scenarios with sliding windows.

8.4.1 *Dataset*

The dataset contains $\approx$ 508,000 observations of payment volumes settled in TIPS for the Swedish krona (SEK) currency, sampled at 15-

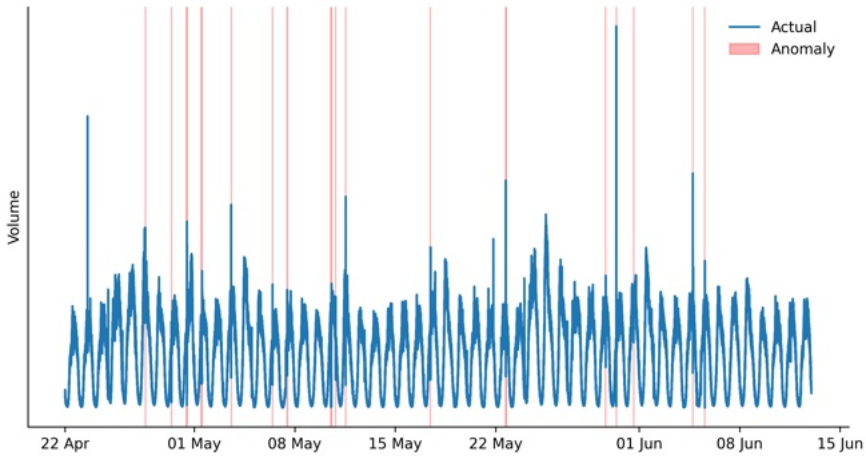


Figure 8.3: Volumes settled in TIPS over time for SEK currency in the test set, with known anomaly intervals highlighted.

second intervals over 89 days (Figure 8.3). This time series presents missing data due to maintenance activities at Swish [135].

We split the dataset into development and test sets: the first 37 days comprised the development set, while the remaining days formed the test set. From the development set, the first 30 days were used for training, with the remainder serving as validation.

This dataset is significant because some anomalies resulted from publicly reported disturbances affecting major Swedish banks, providing ground truth for model evaluation [135]. We considered only anomalies affecting at least 20 transactions per second. The validation set contains 1 anomalous window with 4 anomalous points. The test set contains 19 anomalous windows, totaling 602 anomalous points ($\approx$ 0.20% of the test set).

8.4.2 *Evaluation Criteria*

We consider both threshold-free metrics–AUC-PR, AUC-ROC, PATE, VUS-PR, VUS-ROC, Precision@k–that measure discriminative ability between anomalous and normal observations, and threshold-dependent metrics–MCC, range-based metrics ($\beta = 1$, front bias),

Model	Type	References
LSTM	Deep learning	[165]
Prophet	Statistical	[397]
TBATS	Statistical	[93]
TSM	Deep learning	[62]
TiDE	Deep learning	[91]

Table 8.1: Baseline models compared with the proposed MoGE architectures.

and affiliation–that measure classification performance in decision settings. Additionally, we evaluate functional metrics comparing operational performance: false alarm rate (FAR), event detection rate (EDR), and mean time to detect (MTTD) (for detected events). Metric definitions are provided in Section 2.5.2.

8.4.3 *Baselines*

We compare the proposed MoGE models against five baseline methods comprising both statistical and deep learning approaches (Table 8.1): Time-Series Mixer (TSM) [62], Time-series Dense Encoder (TiDE) [91], Prophet [397], TBATS [93], and long short-term memory (LSTM) [165].

For all models, iterative autoregressive forecasting was performed across the entire test set without being informed of or retrained on newly arriving data. For TSM, TiDE, and LSTM models, we also evaluated predictions using a sliding window approach that simulates real-time deployment, where the model generates forecasts at each time step using a fixed-size window of recent observations. In this scenario, as new observations arrive, the window slides forward (removing the oldest observation and adding the newest), and forecasts are generated without retraining.

8.4.4 *Anomaly Detection*

The anomaly score for an incoming observation is computed by measuring the difference between predicted $\hat{x}_t$ and actual value x_t only when actual values are lower than predicted:

$$s(x_t, \hat{x}_t) = \begin{cases} \hat{x}_t - x_t & \text{if } x_t < \hat{x}_t \\ 0 & \text{otherwise,} \end{cases} \tag{8.5}$$

where $s(\cdot)$ denotes the anomaly score.

This scoring approach specifically targets drops in payment volumes, which are our primary concern. We assign zero score when observed values exceed expected values, which typically occur during normal usage spikes, helping to handle random fluctuations while focusing on payment drops indicating potential issues.

To determine decision thresholds, we adopt two standard criteria computed on a separate validation set. The first is the F1-optimal threshold, defined as

$$t^*_{F_1} = \arg\max_t F_1(t), \tag{8.6}$$

which balances precision and recall and is appropriate when false positives and false negatives have comparable cost. The second is Youden's J statistic [460], defined as

$$t^*_J = \arg\max_t \left[\mathrm{TPR}(t) - \mathrm{FPR}(t)\right], \tag{8.7}$$

which identifies the threshold maximizing overall separability between normal and anomalous instances in terms of TPR and FPR.

8.4.5 *Implementation Details*

We instantiate two implementations of our MoGE architecture: MoGE-cGANs uses conditional GAN as experts, and MoGE-cVAE instantiates conditional VAE models as experts. Both implementations use the same hard gating mechanism and incorporate HTE but differ in generative model structure. Both models use a smoothing factor of 300 for predictions and were implemented in TensorFlow.

For MoGE-cVAE experts, both encoder and decoder employed fully connected layers with dimensions [256, 128] and [128, 256] respectively, using ReLU activation. We optimized the ELBO with mean squared error reconstruction loss and Kullback-Leibler (KL) divergence regularization. To prevent posterior collapse, we implemented KL annealing with a warmup period of 50 epochs and maximum KL weight of 0.5. Training used the Adam optimizer with learning rate 5×10^{-4}, gradient clipping at norm 1.0, and batch size 32 for 300 epochs. The model incorporated a delayed warmup strategy, maintaining zero KL weight for the first 10 epochs before linear annealing. Free bits regularization (0.5 bits per latent dimension) was applied to encourage meaningful latent representations.

The generator architecture for MoGE-cGANs experts consisted of two hidden layers of 64 units each with ReLU activation, while the discriminator employed three hidden layers [256, 128, 64] with ReLU activation and dropout (rate 0.3) for regularization. Adversarial networks were trained using binary cross-entropy loss. We employed the Adam optimizer with learning rate 10^{-4} for both networks, training for up to 1000 epochs with batch size 32 and early stopping.

Baseline methods were implemented using Darts [159]. All models incorporated cyclic day-of-week encoding as future covariates to capture weekly seasonality patterns. For sliding window forecasting models, we evaluated three temporal window configurations: (i) balanced 1-day lookback with 1-day ahead forecast (5760 input/output steps), (ii) long-context short-horizon prediction using 1-day lookback for 5-minute ahead forecasts (5760 input, 20 output steps), and (iii) short-term autoregressive forecasting with 20-step windows for both input and output.

For LSTM models, we used a single-layer architecture with 25 hidden units and no dropout. TiDE models employed a single-layer encoder-decoder structure with hidden dimension 128, temporal attention width 4 for both past and future contexts, and dropout rate 0.1. TSM used 2 conditional mixer blocks with hidden size 64, feedforward size 64, ReLU activation, and LayerNorm. All deep learning models scaled features via reversible instance normalization. Training employed the Adam optimizer with initial learning rate 10^{-4} and exponential decay ($\gamma = 0.999$) to ensure stable convergence. All

models were trained with early stopping monitoring validation loss (patience 10, minimum improvement 10^{-5}) and gradient clipping at norm 1.0. Standard training runs used up to 200 epochs with batch size 256.

8.5 RESULTS

Machine learning forecasting models are evaluated under two distinct strategies. In the iterative autoregressive approach, the model produces multi-step forecasts by repeatedly feeding its own predictions back as inputs, allowing long-horizon forecasting without requiring new observations. In the sliding window approach, the model uses a fixed-size window of recent actual observations to generate forecasts; as new observations arrive, the window slides forward, and the model produces updated forecasts using the refreshed data. Historical forecasts using the sliding window approach are computed by simulating this strategy over the validation period, sequentially generating independent forecast blocks until no further input data remain. Prophet, TBATS, and the two proposed models (MoGE-cVAE and MoGE-cGANs), which cannot leverage recent observations, are evaluated only on long-term autoregressive forecasting.

8.5.1 *Sliding Window Forecasting Accuracy*

Table 8.2 presents threshold-independent metrics, revealing distinct differences in observation separation performance across architectures and input/output window configurations. Figure 8.4 evaluates the precision of top-ranked detections.

TiDE-1D1D demonstrates the strongest performance across most threshold-independent metrics (AUC-ROC 0.9960, AUC-PR 0.7060, VUS-PR 0.1987, PATE 0.7292) and ranks first in all top-ranked decisions. TSM-1D1D exhibits comparable though slightly inferior performance, while achieving marginally higher VUS-ROC (0.8062). In contrast, LSTM-1D1D proves inadequate with 1-day input and output windows.

Model	AUC-PR	AUC-ROC	PATE	VUS-PR	VUS-ROC
LSTM-1D1D	0.0037	0.5256	0.0047	0.0082	0.3928
LSTM-1D5M	0.1606	0.6249	0.1816	0.0702	0.7285
LSTM-5M5M	0.1536	0.6018	0.1747	0.0719	0.7297
TiDE-1D1D	**0.7060**	**0.9960**	**0.7292**	**0.1987**	0.8051
TiDE-1D5M	0.3697	0.9301	0.3854	0.1797	0.7999
TiDE-5M5M	0.0771	0.5684	0.1309	0.0328	0.7164
TSM-1D1D	0.6147	0.9923	0.6236	0.1824	**0.8062**
TSM-1D5M	0.2024	0.6955	0.2259	0.0779	0.7612
TSM-5M5M	0.0599	0.5720	0.1235	0.0348	0.7183

Table 8.2: Sliding window forecasting threshold-independent metrics for LSTM, TiDE, and TSM models with varying input and output window configurations.

The transition to 5-minute output granularity (1D-5M) produces divergent outcomes: while TiDE experiences substantial degradation in all Precision@k metrics, both LSTM-1D5M and TSM-1D5M maintain competitive Precision@100 scores (0.79 and 0.92 respectively).

Configurations employing 5-minute input and output windows underperform across all architectures, with the exception of LSTM-5M5M, which achieves better precision on the highest-ranked anomalies compared to TiDE-5M5M and TSM-5M5M using the same input and output windows.

Table 8.3 presents operational results using both F1-optimal threshold $t^*_{F_1}$ and Youden index t^*_J calculated on the validation set. Models using $t^*_{F_1}$ consistently outperform the Youden index, which produces impractically high false alarm rates (3.5-36 FAR/h) and poor precision (<1%).

Considering individual anomalous observations (points) and using the F1-based threshold, TiDE-1D1D achieves the highest overall performance with MCC 0.6632, followed by TSM-1D1D (MCC 0.6314). However, these models fail to detect all anomaly events (EDR < 1), whereas all models with 1D5M and 5M5M input/output windows achieve complete event detection. Among these, LSTM-1D5M

		LSTM			TiDE			TSM		
	Metric	1D1D	1D5M	5M5M	1D1D	1D5M	5M5M	1D1D	1D5M	5M5M
$t^*_{F_1}$	MCC	0.0418	0.2584	0.3094	**0.6632**	0.4957	0.2456	0.6314	0.3488	0.2348
	Precision_T	0.0186	0.1635	**0.4264**	0.2670	0.2664	0.2427	0.2970	0.2247	0.2206
	Recall_T	0.0020	0.8170	0.8384	0.2944	0.8332	0.7020	0.3087	**0.8737**	0.7262
	F_{1_T}	0.0037	0.2725	**0.5653**	0.2800	0.4037	0.3607	0.3027	0.3574	0.3384
	$\text{P}_{\text{precision}}$	0.4152	0.7995	0.8696	0.7511	0.8117	0.8658	0.7627	0.8058	**0.8913**
	P_{recall}	0.2818	0.9975	0.9973	0.6802	**0.9999**	0.9977	0.6582	0.9975	0.9978
	P_{F_1}	0.3357	0.8876	0.9291	0.7139	0.8960	0.9271	0.7066	0.8915	**0.9416**
	FAR/h	5.2619	0.1279	**0.0306**	0.1351	0.2533	0.0869	0.1343	0.0508	0.1054
	EDR	0.0526	**1.0000**	**1.0000**	0.5263	**1.0000**	**1.0000**	0.5789	**1.0000**	**1.0000**
	MTTD_{obs} *	22.0000	**0.1053**	0.3158	1.7000	0.3684	0.4211	2.0000	0.3158	0.4211
t^*_J	MCC	0.0412	0.0631	0.0775	0.2250	0.1698	0.0460	**0.2552**	0.0382	0.0507
	Precision_T	**0.0066**	0.0038	0.0065	0.0029	0.0041	0.0047	0.0035	0.0018	0.0056
	Recall_T	0.0055	**0.9093**	0.8992	0.8917	0.9061	0.7833	0.8627	0.8982	0.7796
	F_{1_T}	0.0060	0.0076	**0.0129**	0.0057	0.0082	0.0094	0.0070	0.0037	0.0111
	$\text{P}_{\text{precision}}$	0.4541	0.5679	0.5749	0.5290	0.5552	0.5722	0.5174	0.5398	**0.5831**
	P_{recall}	0.4163	0.9988	0.9988	**1.0000**	1.0000	0.9994	1.0000	0.9998	0.9994
	P_{F_1}	0.4343	0.7240	0.7297	0.6920	0.7140	0.7277	0.6819	0.7011	**0.7365**
	FAR/h	7.4446	5.2950	**3.4838**	8.4025	11.5439	7.4034	6.0428	36.2331	5.9897
	EDR	0.0526	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**	**1.0000**
	MTTD_{obs} *	16.0000	**0.0000**	**0.0000**	0.1053	**0.0000**	**0.0000**	0.4211	**0.0000**	0.0526

* MTTD is computed only over detected events. EDR indicates the fraction of anomaly events detected. When EDR = 0, MTTD is undefined.

Table 8.3: Sliding window forecasting threshold-dependent metrics using F1-optimal threshold ($t^*_{F_1}$) and Youden index (t^*_J) for LSTM, TiDE, and TSM models.

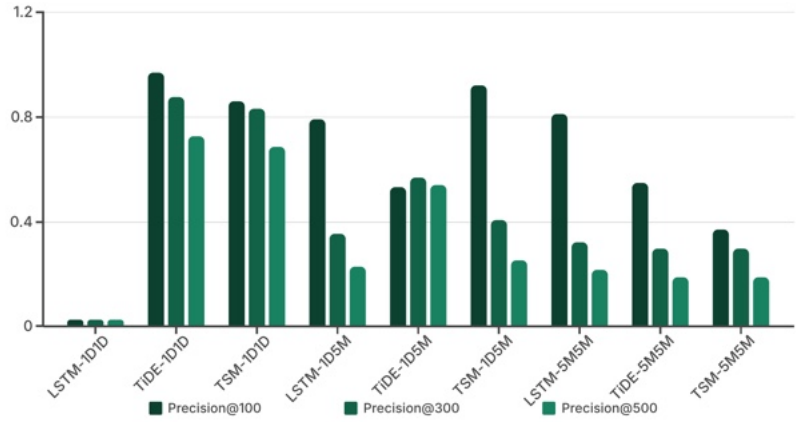


Figure 8.4: Proportion of true anomalies among top-k observations ranked by anomaly score for sliding window forecasting models, evaluated at $k \in \{100, 300, 500\}$.

achieves the lowest MTTD, while LSTM-5M5M achieves the lowest FAR/h (0.0306).

These findings suggest that models with better theoretical separability do not necessarily correspond to superior operational performance when threshold-based decisions are required.

8.5.2 *Iterative Autoregressive Forecasting Performance*

Long-term iterative autoregressive forecasting is evaluated for LSTM, TiDE, and TSM models (using previous outputs as inputs without waiting for new real-time observations), alongside Prophet, TBATS, and the two proposed models, MoGE-cVAE and MoGE-cGANs.

Figure 8.5 shows that the ranking capability for classifying observations degrades substantially for all LSTM and TSM variants. AutoTBATS also fails to manage high-frequency data and seasonal complexity over time.

Figure 8.6 shows that these models degrade over time because they accumulate errors (as with Prophet or TSM) or fail to capture seasonality and trend (as with LSTM in the worst cases).

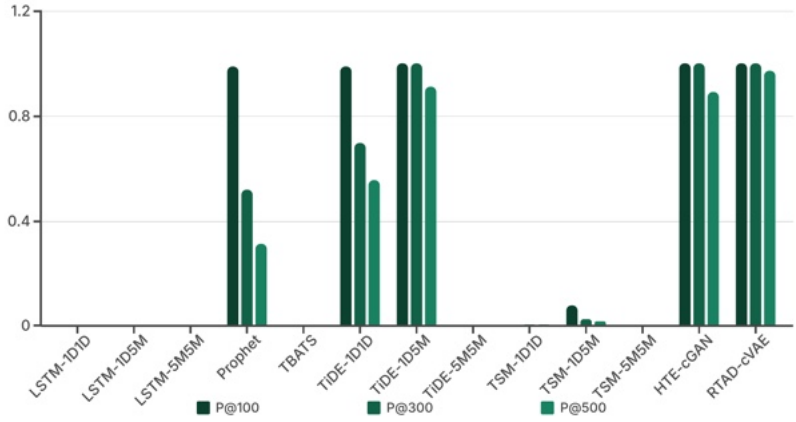


Figure 8.5: Proportion of true anomalies among top-k observations ranked by anomaly score for iterative autoregressive forecasting models, evaluated at $k \in \{100, 300, 500\}$.

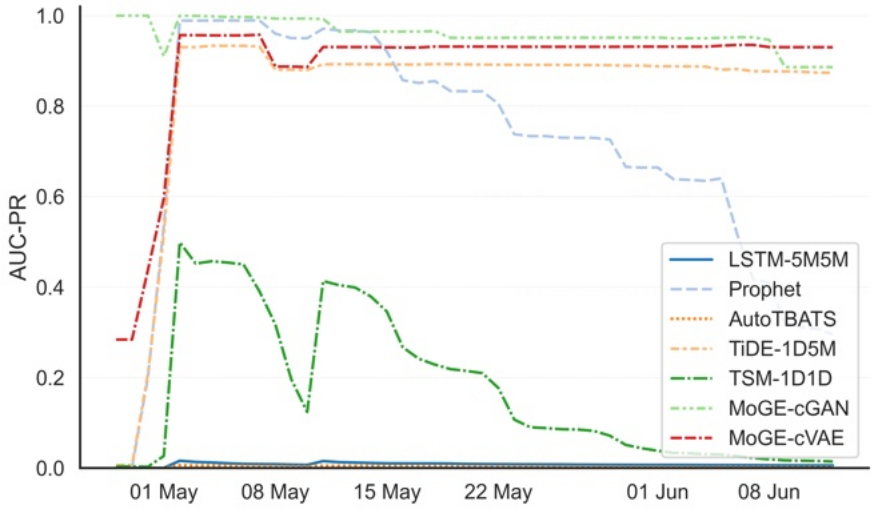


Figure 8.6: AUC-PR calculated in streaming mode over time, day by day, for long-term iterative autoregressive forecasting models.

Model	AUC-PR	VUS-PR	AUC-ROC	VUS-ROC	PATE
LSTM-1D1D	0.0020	0.0074	0.4932	0.4890	0.0029
LSTM-1D5M	0.0020	0.0075	0.4956	0.4950	0.0028
LSTM-5M5M	0.0056	0.0081	0.7832	0.4558	0.0056
Prophet	0.2746	0.0457	0.9188	0.7281	0.2855
TBATS	0.0020	**0.5071**	0.5000	0.5031	0.5023
TiDE-1D1D	0.5272	0.1058	0.9698	0.5877	0.5355
TiDE-1D5M	0.8731	0.1963	0.9939	0.6022	0.9010
TiDE-5M5M	0.0130	0.0192	0.6065	0.4847	0.0176
TSM-1D1D	0.0129	0.0369	0.9174	**0.8837**	0.0179
TSM-1D5M	0.0081	0.0092	0.7246	0.4313	0.0088
TSM-5M5M	0.0027	0.0154	0.5507	0.4793	0.0053
MoGE-cGAN	0.8861	0.2589	0.9995	0.8601	0.8967
MoGE-cVAE	**0.9300**	0.2518	**0.9996**	0.7178	**0.9314**

Table 8.4: Iterative autoregressive forecasting threshold-independent metrics for all evaluated models.

Table 8.4 shows that MoGE-cVAE achieves the best performance in terms of AUC-PR, AUC-ROC, and PATE, followed by MoGE-cGANs and TiDE-1D5M. This ranking persists when models are deployed with static thresholds, as shown in Table 8.5. MoGE-cVAE emerges as the best overall model, followed by MoGE-cGANs.

Comparing with Table 8.3, MoGE-cVAE achieves performance competitive with or superior to sliding window models that have access to more recent observations. While access to recent data might help models adapt to new changes in real-time, in this context it exposes them to strong oscillations in payment volumes that disturb predictions, an issue from which MoGE models are immune, similar to Prophet. However, unlike Prophet, predictions accumulate fewer errors over time, resulting in a more robust baseline.

8.5.3 *Resource Utilization*

Deep learning models were trained with GPUs, but detection operations were deployed on virtual machines (VMs) with limited re-

	Metric	LSTM	Prophet	TiDE	TSM	MoGE-cGAN	MoGE-cVAE
$t^*_{F_1}$	MCC	0.0315	0.4921	0.8312	0.0498	0.8508	**0.8874**
	Precision_T	0.0113	0.8000	0.5326	0.0096	**1.0000**	0.7419
	Recall_T	0.0009	0.3365	0.5280	0.2934	0.5686	**0.7959**
	F_{1_T}	0.0017	0.4737	0.5303	0.0186	0.7250	**0.7680**
	$\text{P}_{\text{precision}}$	0.3757	0.9792	0.9898	0.6723	**1.0000**	0.9873
	P_{recall}	0.4934	0.3684	0.7894	0.6232	0.6840	**0.9473**
	P_{F_1}	0.4266	0.5353	0.8783	0.6468	0.8123	**0.9669**
	FAR/h	4.7312	0.0016	0.0370	5.8024	**0.0000**	0.0088
	EDR	0.0526	0.3684	0.7895	0.5263	0.6842	**0.9474**
	$\text{MTTD}_{obs}{}^*$	16.0000	2.0000	1.6667	25.3000	8.6923	2.0556
t^*_J	MCC	0.0490	0.0670	0.3337	0.0811	**0.4578**	0.4008
	Precision_T	0.0032	0.0044	0.0147	0.0013	0.0187	0.0109
	Recall_T	0.7472	0.8713	0.8531	0.8163	0.9305	**0.9681**
	F_{1_T}	0.0063	0.0088	0.0290	0.0026	**0.0366**	0.0216
	$\text{P}_{\text{precision}}$	0.4659	0.7523	0.8094	0.5989	**0.8582**	0.8210
	P_{recall}	0.9319	0.9474	0.9261	1.0000	1.0000	**1.0000**
	P_{F_1}	0.6212	0.8386	0.8638	0.7492	**0.9237**	0.9017
	FAR/h	101.3589	68.2206	3.3905	52.1425	1.7878	2.4713
	EDR	0.8421	0.9474	0.8947	0.9474	**1.0000**	**1.0000**
	$\text{MTTD}_{obs}{}^*$	0.2500	0.1667	0.3529	0.0556	0.1579	0.0526

* MTTD is computed only over detected events. EDR indicates the fraction of anomaly events detected. When EDR = 0, MTTD is undefined.

Table 8.5: Iterative autoregressive forecasting threshold-dependent metrics. For brevity, only the best configurations for LSTM, TiDE, and TSM are shown (LSTM-5M5M, TiDE-1D5M, TSM-1D1D).

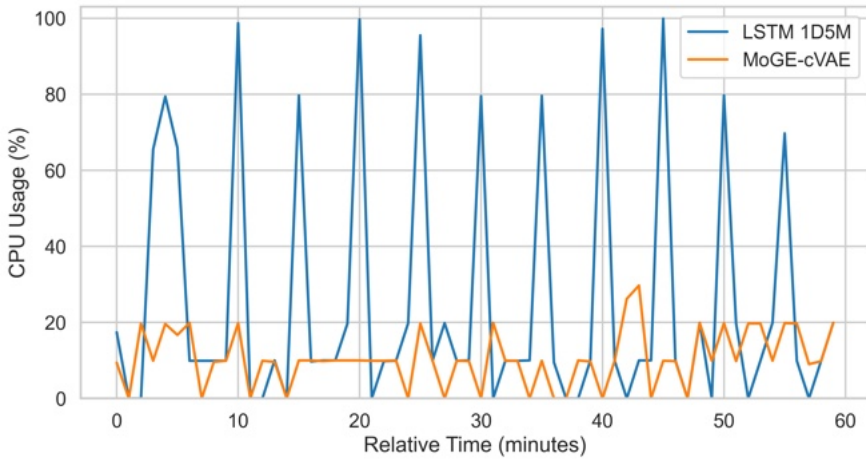


Figure 8.7: CPU usage per minute during inference on the first 384 observations (1 hour at 15-second sampling frequency) on a VM with 1 CPU, 2GB RAM, and no GPU.

sources (1 CPU, 2GB RAM, no GPU) to evaluate performance under realistic constraints typical of production environments.

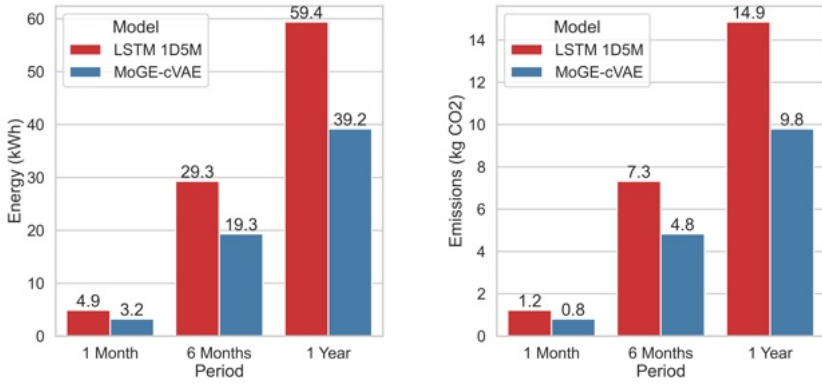
Figure 8.7 shows CPU usage over the first 384 observations (1 hour with 15-second sampling frequency) comparing one of the smaller models (LSTM-1D5M) that fits in memory against the MoGE-cVAE model.

Figure 8.8 presents the energy consumption and emissions footprint during inference, comparing iterative and sliding window forecasting models.

Table 8.6 presents model sizes and parameter counts. The MoGE architectures maintain relatively compact sizes, with MoGE-cGANs at 295 KB (32,725 total parameters across 7 experts) and MoGE-cVAE at 1.089 MB (241,045 total parameters across 7 experts).

8.5.4 *Qualitative Analysis*

Figures 8.9 through 8.14 present a qualitative comparison of forecast outputs across different model configurations. The visualizations



(a) Energy consumption

(b) Emissions

Figure 8.8: Environmental footprint of models on the univariate SEK dataset in terms of energy consumption and emissions during inference, comparing iterative autoregressive and sliding window forecasting approaches.

reveal distinct behavioral patterns regarding how each model handles seasonal structures, trends, and anomalous observations.

MoGE-cVAE (Figure 8.9) and MoGE-cGANs (Figure 8.10) both successfully capture the fundamental seasonal structure and maintain consistent amplitude throughout the forecast horizon. MoGE-cGANs produces sharper predictions with higher variability, whereas MoGE-cVAE generates slightly smoother outputs that better represent the central tendency of the data distribution.

Sliding window forecasting models adapt rapidly to incoming observations. For instance, as illustrated in Figure 8.11, the LSTM-1D5M model incorporates anomalous data into its input window, causing the forecast to track the anomaly rather than maintaining a stable baseline. While this behavior initially allows for the detection of the anomaly's onset, it compromises the integrity of anomaly scoring, as the residual error remains low even during significant deviations, thereby achieving low point-based error metrics. Furthermore, this adaptation obscures the determination of when the anomaly has concluded and when normal conditions have been restored.

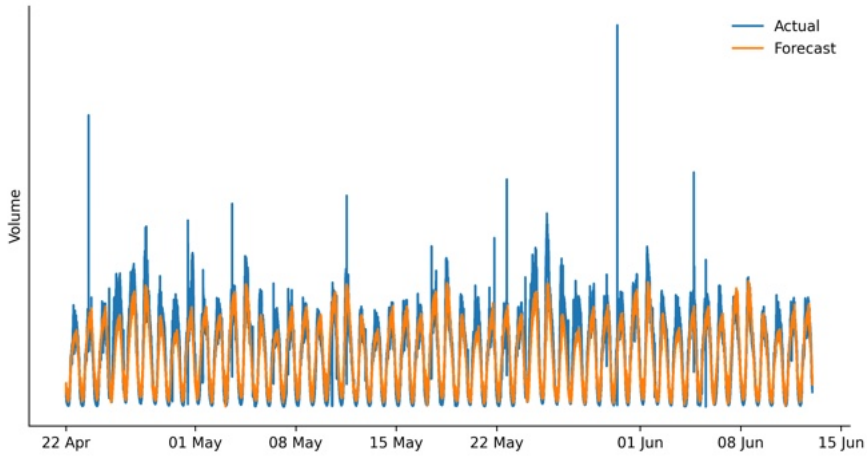


Figure 8.9: Comparison of actual and predicted time series with detected anomalies using MoGE-cVAE. The model maintains a stable reconstruction of the seasonal pattern.

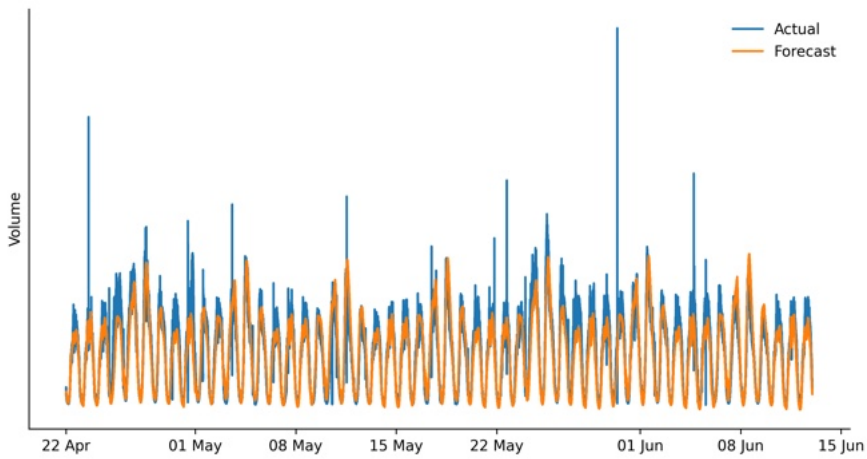


Figure 8.10: Comparison of actual and predicted time series with detected anomalies using MoGE-cGANs, showing sharper variability in the forecast.

Model Name	Total Parameters	Model Size (MB)
AutoTBATS	–	115.521
LSTM-1D1D	3,442	0.014
LSTM-1D5M	3,442	0.014
LSTM-5M5M	3,442	0.014
Prophet	–	23.965
TSM-1D1D	104,750,355	419.001
TSM-1D5M	173,295	0.693
TSM-5M5M	58,495	0.234
TiDE-1D1D	981,650,248	3926.601
TiDE-1D5M	10,803,868	43.215
TiDE-5M5M	1,505,068	6.020
MoGE-cGAN [1]	32,725	0.295
MoGE-cVAE [2]	241,045	1.089

[1] 7 experts, each with 4,675 params ($\approx$ 43 kB)
[2] 7 experts, each with 34,435 params ($\approx$ 159 kB)

Table 8.6: Parameter counts and model sizes for the evaluated anomaly detection architectures.

In contrast, other models performing long-term forecasting without updates from new observations exhibit performance that degrades differently over time. Prophet (Figure 8.12) suffers from significant trend divergence. While the model accurately captures the seasonal periodicity, the extrapolation of the estimated trend leads to a distinct upward drift in the baseline. Consequently, the forecast deviates substantially from the actual volume levels over the long horizon, resulting in periods of consistent overestimation.

TSM, when operating with 1-day input and output windows in iterative mode (Figure 8.13), suffers from rapid accumulation of error after the initial days. Rather than maintaining a stable structure, the model exhibits distinct seasonal instability, characterized by a progressive amplification of the wave amplitude. This distortion

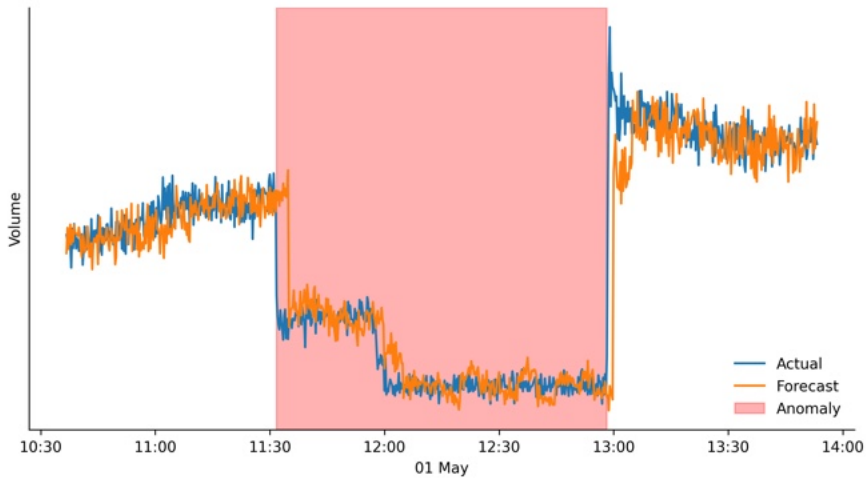


Figure 8.11: Sliding window forecasting with LSTM using 1-day input and 5-minute output windows (LSTM-1D5M). The forecast closely tracks the anomaly (red zone), demonstrating excessive adaptation to recent inputs.

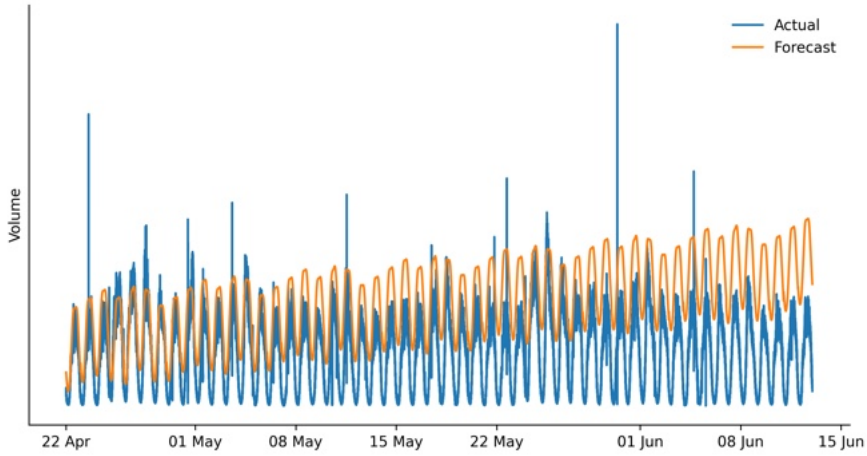


Figure 8.12: Long-term forecasting with Prophet, showing baseline drift and trend divergence resulting in overestimation.

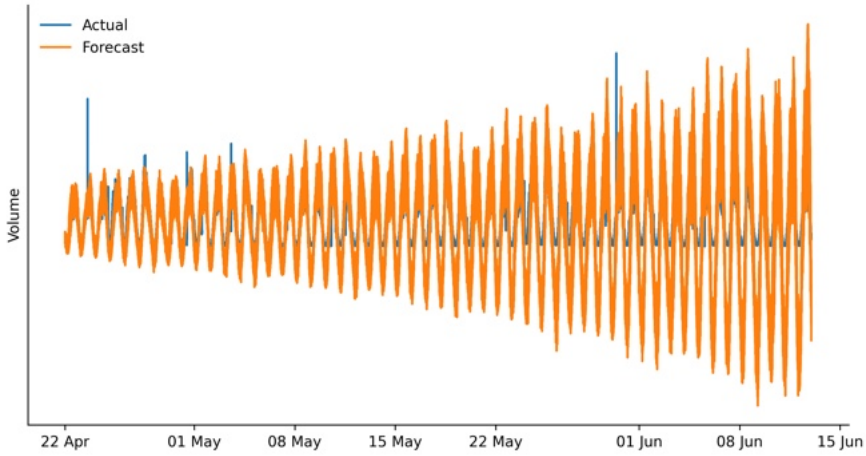


Figure 8.13: Long-term autoregressive forecasting with TSM using 1-day input and output windows (TSM-1D1D). The model fails to maintain stable variance, resulting in erroneous seasonal amplification.

leads to exaggerated peaks and troughs that do not reflect the actual data distribution, resulting in significant alternating periods of overestimation and underestimation.

Conversely, the TSM model utilizing 1-day input and 5-minute output windows, also operating in autoregressive mode (Figure 8.14), exhibits severe signal degradation. The model produces overly smooth predictions that rapidly collapse toward the mean, failing to sustain any meaningful seasonal or intra-day variations.

8.6 DISCUSSION

The experimental results provide insights into the performance characteristics, operational trade-offs, and limitations of the proposed MoGE architecture for streaming anomaly detection in high-frequency payment systems.

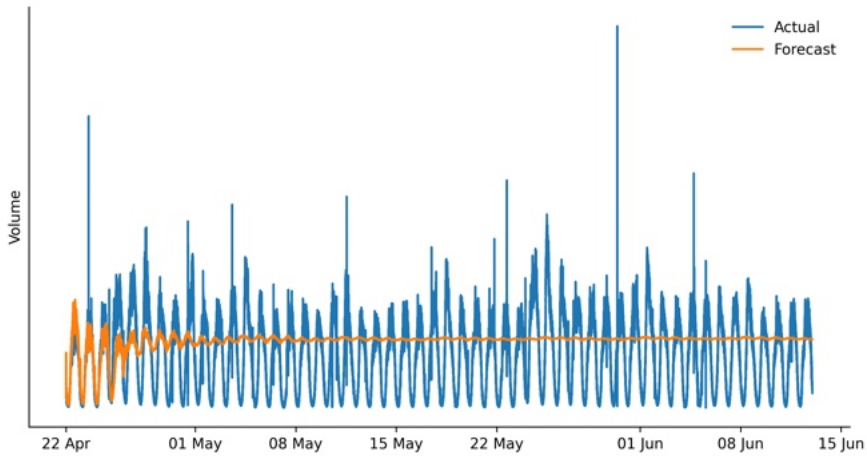


Figure 8.14: Long-term autoregressive forecasting with TSM using 1-day input and 5-minute output windows (TSM-1D5M), showing severe amplitude attenuation and a complete loss of temporal structure.

8.6.1 *Forecast Horizon and Anomaly Detection Effectiveness*

The experiments reveal that the forecast horizon length affects anomaly detection capabilities. Models with excessively short forecast horizons (e.g., 5 minutes) tend to adapt too readily to incoming patterns, including anomalous ones, making them ineffective for detection. This behavior is evident in Figure 8.11, where the sliding window forecasting model adjusted its predictions to match the anomalous drop rather than maintaining a stable baseline.

Conversely, models employing iterative autoregressive forecasting over extended periods accumulate noise and error, although they maintain better resistance to anomaly adaptation. Specifically, TSM exhibits seasonal instability and variance amplification (Figure 8.13), while Prophet suffers from trend divergence (Figure 8.12). The most stable configuration among baselines utilizes a 1-day historical context, which offers a balance between temporal stability and anomaly sensitivity. However, this configuration requires a complete day of historical data for future predictions, representing a trade-off be-

tween accuracy and data dependency that may not be acceptable in all operational contexts.

The MoGE models circumvent this trade-off by generating forecasts directly from timestamp-derived features without requiring recent observations. This enables stable anomaly detection without the risk of adapting to anomalous patterns while avoiding the error accumulation and variance instability associated with iterative approaches.

8.6.2 *Comparative Performance Analysis*

MoGE-cVAE achieves the strongest overall performance in long-term forecasting scenarios with no access to recent observations, yielding an AUC-PR of 0.9300, an AUC-ROC of 0.9996, and a PATE of 0.9314. MoGE-cGANs demonstrates competitive performance with an AUC-PR of 0.8861 and an AUC-ROC of 0.9995. Both models substantially outperform baseline methods in threshold-independent metrics, with TiDE-1D5M representing the closest competitor among traditional architectures (AUC-PR 0.8731, AUC-ROC 0.9939).

The performance advantage of MoGE models persists even when compared to sliding window forecasting models that had access to recent observations. While long-term forecasting without recent data reduces the risk of adapting to anomalous patterns, it typically increases sensitivity to noise. However, models with access to recent observations in this high-frequency payment context are exposed to strong oscillations that disturb predictions. The MoGE architecture demonstrates resilience to both upward spikes and noise fluctuations, providing immunity to these disturbances while maintaining lower error accumulation over time compared to both sliding window approaches and Prophet.

Given that transaction volume is represented through temporal bucketing, where each observation aggregates multiple transactions, point-based metrics provide a meaningful evaluation for streaming anomaly detection in this context. The observed performance differences in point-based metrics, therefore, reflect relevant operational characteristics for the deployment scenario.

8.6.3 *Sensitivity to Window and Threshold Configuration*

The experiments reveal that architectural performance is highly sensitive to input and output window configurations. The same architecture exhibits substantially different behaviors depending on window parameters, as demonstrated by the performance variation observed across LSTM configurations. This window sensitivity presents an operational challenge, as optimal configurations are dataset-dependent and require extensive hyperparameter searches. The MoGE architecture addresses this challenge by operating without sliding windows, requiring only the specification of the primary seasonal period, which is typically known a priori in seasonal forecasting applications.

The evaluation demonstrates that threshold-independent metrics and threshold-dependent operational metrics capture different aspects of model performance. In operational deployment, models require specific thresholds to function in real-time anomaly detection, and the impact of threshold selection becomes evident in production settings. Models with superior ranking capability do not necessarily translate to superior operational performance when threshold-based decisions are required.

The MoGE-based models demonstrate robustness to threshold selection, maintaining performance when thresholds calculated on the validation set were applied to the test set. Among autoregressive baseline models, only TiDE-1D1D achieves comparable stability under operational threshold constraints, while other architectures show greater performance degradation when transitioning from ranking-based evaluation to threshold-based deployment.

8.6.4 *Operational Deployment Considerations*

The resource requirements of MoGE implementations support their deployment in resource-constrained environments, including edge devices. Table 8.6 shows that MoGE-cGANs and MoGE-cVAE utilize 32,725 (295 KB) and 241,045 (1.089 MB) parameters, respectively. These storage requirements are significantly lower than those of dense encoder baselines such as TiDE-1D1D, which exceeds 3.9 GB.

Figure 8.7 indicates that MoGE-cVAE maintains stable CPU usage during inference on a standard virtual machine (1 CPU, 2 GB RAM), in contrast to the recurrent utilization spikes observed in the sliding window LSTM-1D5M model. The reduced computational requirements translate to lower energy consumption. As illustrated in Figure 8.8, MoGE-cVAE exhibits lower cumulative energy consumption and carbon emissions compared to the LSTM baseline over projected 6-month and 1-year periods.

This energy differential is substantial. In multivariate scenarios, the efficiency advantages of generative approaches over sliding window methods would be more pronounced, as demonstrated in Chapter 7 where experiments on 16-dimensional dataset revealed a 1255-fold reduction in energy consumption.

Unlike traditional MoE architectures that activate multiple experts and require complex soft-gating mechanisms, the MoGE framework employs rule-based hard-gating driven by calendar and temporal features. This design choice provides deterministic expert selection and explicit traceability, which are operational requirements in financial compliance contexts where model decisions must be auditable. The temporal encoding's effectiveness is demonstrated through the model's ability to maintain coherence across different experts, enabling the modular architecture to assign expert responsibilities based on temporal characteristics without learned routing mechanisms.

The generative architecture enables predictions to be produced without immediate dependence on recent observations, which allows operational continuity during periods of data unavailability or transmission interruptions. Furthermore, the modular expert structure allows for selective retraining in response to drift, reducing operational maintenance requirements compared to monolithic models.

8.6.5 *Generative Model Comparison*

The comparison between MoGE-cVAE and MoGE-cGANs reveals complementary strengths. GAN-based experts produce sharper, more realistic predictions that better capture fine-grained temporal variations. However, this characteristic can be disadvantageous in noisy

high-frequency contexts, where overly detailed reconstruction may reduce anomaly detection sensitivity by modeling noise as signal.

VAE-based experts produce smoother reconstructions that better represent distributional central tendencies. While this results in reduced fidelity to individual observations, it provides a more stable baseline for anomaly detection in high-frequency time series with substantial noise. The smoothing effect effectively filters normal variability while maintaining sensitivity to significant deviations, resulting in superior overall anomaly detection performance as evidenced by the experimental results.

8.6.6 *Limitations*

The MoGE architecture does not incorporate online learning mechanisms. When temporal patterns evolve due to concept drift, the affected experts require retraining. While selective expert retraining is more efficient than updating an entire monolithic model, it still necessitates periodic model updates and cannot adapt continuously to gradual pattern evolution. The rule-based hard-gating mechanism, while providing deterministic and auditable expert selection, lacks the adaptability of learned soft-gating approaches that could potentially adjust expert routing based on emerging patterns. Applications requiring continuous adaptation to evolving distributions may benefit from hybrid approaches that incorporate online learning mechanisms or adaptive gating strategies.

9 FAILURE DETECTION IN INSTANT PAYMENT INFRASTRUCTURES

Traditional monitoring systems often operate in isolated silos, storing metrics separately across different tools and platforms [211, 319]. In distributed systems, this fragmentation prevents holistic infrastructure visibility, as individual component alerts fail to reveal system-level issues, complicating anomaly detection and prolonging incident response times. This chapter presents a failure detection framework built on GenTT that addresses these observability gaps (RQ6). We extract meaningful features from the SCT Inst process based on ISO 20022 messages and formulate failure detection as an anomaly detection problem on these engineered features. This approach enables early detection of issues both within and external to the distributed infrastructure while providing operators with actionable explanations to assess business impact and localize failures in real-time, bridging the gap between technical metrics and business processes.

9.1 INTRODUCTION

The evolution toward real-time payment processing is transforming both modern economies and payment infrastructure management. Unlike traditional batch-processing systems, instant payments operate continuously without business days, available around the clock. Each transaction settles within seconds. A malfunction lasting just a few seconds in a CSM could immediately affect hundreds of customers, as no buffer period exists for addressing issues before transaction completion.

To ensure quality of service with zero-downtime expectations, modern RTGS systems such as TIPS rely on distributed architectures [11, 56]. While offering benefits in scalability, availability, and reliability, this decentralized design introduces operational complexity com-

pared to traditional RTGS systems through increased abstraction layers and component interdependencies. When failures occur, localizing root causes and assessing business impact across the distributed infrastructure becomes critical.

Operational management solutions relying on manual intervention by IT operators are no longer feasible at this scale. AIOps [224] solutions reduce manual effort in monitoring, anomaly detection, root cause analysis, and recovery. However, their effectiveness depends on data quality and domain-specific adaptation. Monitoring solutions collecting telemetry data often fail to capture semantic relationships between technical metrics and business processes, hindering accurate business impact assessment of technical anomalies. This observability gap is particularly evident in financial payment infrastructures where dependencies extend across organizational boundaries, delaying MTTR due to difficulties in localizing root causes within the broader settlement process.

We present a failure detection framework based on anomaly detection applied to processing times between consecutive ISO 20022 message exchanges. By mapping these temporal features to distinct processing phases, we create a compact representation of the system state that bridges infrastructure observability and business process visibility. The framework treats detected anomalies as indicators of potential failures, employing an explainability-by-design approach that integrates domain knowledge to enable failure localization, incident classification, and business impact estimation. This approach complements traditional monitoring systems by detecting service degradation invisible at the component level, differentiating between internal and external issues, and delivering actionable explanations that guide remediation before escalation.

The remainder of this chapter is organized as follows. Section 9.2 describes the SCT Inst scheme and the ISO 20022 message flow from which temporal features are computed. Section 9.3 formalizes the failure detection problem and presents the proposed framework. Section 9.4 describes the real-time feature extraction architecture, experimental testbed design, datasets, evaluation metrics, and implementation details. Section 9.5 reports experimental findings from both testbed scenarios and a real-world network service provider

(NSP) incident, while Section 9.6 discusses advantages, limitations, and potential future developments.

9.2 THE SEPA INSTANT CREDIT TRANSFER SCHEME

The SEPA Instant Credit Transfer (SCT Inst) scheme [117] represents a significant harmonization effort for instant payment systems across Europe. This scheme establishes a target maximum execution time of 5 seconds and imposes stringent availability requirements, as services adhering to SCT Inst rules must be available 24/7 on all calendar days.

An SCT Inst transaction directly involves four key parties: the payer who initiates the transfer (Originator), the payer's PSP that processes the instruction (Originator PSP), the recipient's PSP that credits the funds (Beneficiary PSP), and the recipient who receives the payment (Beneficiary).

These transactions are facilitated through CSMs, comprising one or more entities performing clearing and settlement functions. Clearing encompasses information exchange and determination of final settlement positions, while settlement involves extinguishing obligations defined during clearing. The two principal CSM organizations compliant with the SCT Inst scheme are RT1 [102] and TARGET Instant Payment Settlement (TIPS) [86, 335].

The maximum execution time of 5 seconds is measured from the moment the Originator PSP receives the instruction until it receives confirmation (positive or negative) of settlement. If the Originator PSP has not received any confirmation message within 10 seconds of the timestamp, it must immediately restore the Originator's account by lifting the reservation of the amount.

In a typical scenario where two PSPs participate in the same CSM, the SCT Inst settlement process begins when the Originator requests the Originator PSP (typically a commercial bank) to execute an instant payment. After reserving the transfer amount in the Originator's account, the Originator PSP transmits the order to the CSM, which dispatches the instruction to the Beneficiary PSP and awaits acceptance or rejection.

The Beneficiary PSP performs a series of checks before sending an acceptance or rejection notification to the CSM, which relays it back to the Originator PSP. If the payment is accepted, the Beneficiary PSP makes the funds available in the Beneficiary's account, and the Originator PSP debits the reserved amount. Conversely, if the payment is rejected, the Originator PSP must inform the Originator of the Beneficiary PSP's decision.

From a technical perspective, each step of the SCT Inst process corresponds to a message following the global standard for financial messaging ISO 20022 [402]. We create new features by computing processing times between relevant message exchanges in the SCT Inst process visible to the CSM. By combining these processing time features with the volume of settled instant payments, we create a compact representation of payment system state. The next section describes the formulation of a failure detection problem leveraging this representation.

9.3 A FAILURE DETECTION FRAMEWORK FOR CLEARING AND SETTLEMENT MECHANISMS

Processing times across SCT Inst phases provide unique observability of distributed components in an instant payment system. Delays in message exchange often indicate infrastructure issues; by coupling processing time information with settled payment volumes, we can represent the holistic operational state of the payment infrastructure, bridging the gap between technical infrastructure metrics and business process visibility.

This section presents a framework for preemptive failure detection in CSM infrastructures. We first formalize the problem as an anomaly detection task with two objectives: identifying irregular system behavior and characterizing its origin and severity. We then show how to construct a compact system state representation from transaction processing times and volumes. The resulting framework combines detection and explanation capabilities to provide IT operators with actionable insights during incident response.

9.3.1 *Problem Formulation*

We define the failure detection problem in a clearing and settlement mechanism (CSM) infrastructure as identifying failures before they escalate into incidents. We approach this as an anomaly detection task, assuming failures manifest as irregular patterns in observable metrics, such as increased processing times or reduced settled payment volumes, indicating the platform's diminished ability to fulfill its core service function.

Formally, let $S^\eta = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_m\}$ denote a multivariate time series of metrics observed at frequency η, where each observation $\mathbf{s}_k \in \mathbb{R}^d$ at time k. An observation $\mathbf{s}_k$ is anomalous if at least one of its d components deviates significantly from expected behavior.

The fundamental challenge is to determine, in near real-time (i.e., at time $k + \varepsilon$, where ε is a negligible time interval), whether the current operational state is anomalous and, if so, to identify the originating infrastructure component and assess potential severity.

This challenge decomposes into two distinct subproblems:

1. *Anomaly detection*: estimating the function $f : \mathbb{R}^d \to \{0, 1\}$ that determines whether an observation $\mathbf{s}_k$ is anomalous, where $A = \{\mathbf{s}_k \in S^\eta : f(\mathbf{s}_k) = 1\}$ denotes the set of all anomalous observations;

2. *Anomaly characterization*: given a detected anomaly $\mathbf{a} \in A$, estimating the function $g : A \to \mathcal{L} \times \mathcal{S}$. For instance, if $\mathcal{L} = \{A_1, A_2\}$, g determines whether $\mathbf{a}$ originates within the CSM infrastructure ($\mathbf{a} \in A_1$) or from external factors ($\mathbf{a} \in A_2$), where $A_1, A_2 \subseteq A$ form a partition of A, and assigns a severity level in $\mathcal{S}$ to assess potential impact.

Upon completing these phases, the system provides explanations for identified anomalies that guide IT operators' investigations by offering insights into the distributed system's state. For instance, A_1 encompasses anomalies caused by misconfigurations or hardware failures within the CSM infrastructure, while A_2 includes anomalies arising from external factors such as network issues or participant-related problems.

9.3.2 *System State Representation*

To enable preemptive failure detection, we assume the existence of a time series S^{η} capturing relevant operational patterns of the instant payment system. Constructing such a representation requires selecting informative features and aggregating them efficiently for real-time processing. This section presents the feature selection rationale and the aggregation methodology transforming irregular transaction data into a regular time series suitable for anomaly detection.

9.3.2.1 *Processing Time Features*

From a technical perspective, an instant payment transaction consists of standardized ISO 20022 messages exchanged between participating entities. The CSM engages twice during processing: first in the conditional phase (validation, fund reservation, and notification to the beneficiary participant) and later in the settlement phase. The relevant messages are pacs.008 and pacs.002, where pacs.008 initiates a request while pacs.002 responds to it.

In the standard settlement model, a transaction can have four possible final states: settled, expired due to timeout, rejected by the beneficiary, or failed due to validation errors or insufficient liquidity. Our framework considers only settled payments, providing sufficient information to establish a baseline model and calculate all processing time features. As illustrated in Figure 9.1, three processing phases are defined: the duration of the conditional phase within the CSM (Phase A), the cumulative time spent outside the CSM awaiting the beneficiary's reply (Phase B), and the time within the CSM for the settlement phase (Phase C).

An instant payment is formally defined as a tuple

$$\mathbf{p} = (\delta_1, \delta_2, \delta_3), \tag{9.1}$$

where:

- δ_1 approximates Phase A duration, computed as the time difference between the CSM sending the pacs.008 message to the beneficiary bank and the CSM receiving the pacs.008 message from the originator bank;

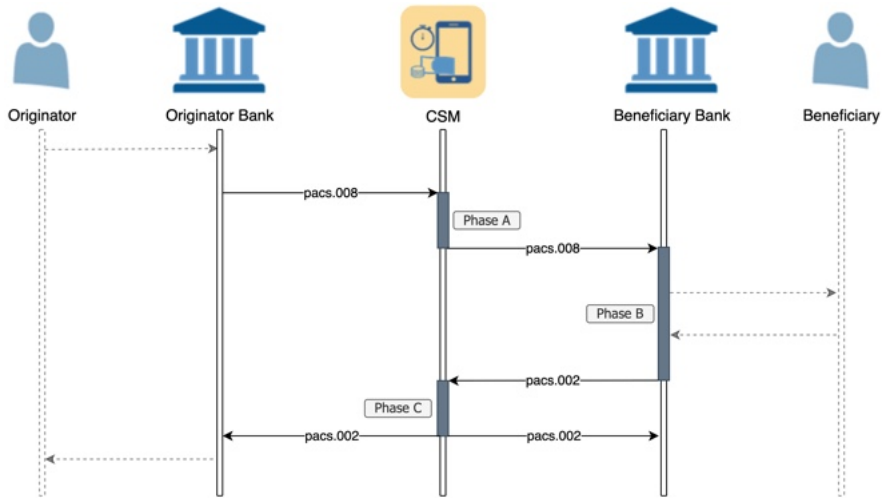


Figure 9.1: Core SCT Inst message flow highlighting the three processing phases whose durations serve as features for anomaly detection.

- δ_2 approximates Phase B duration, computed as the time difference between the CSM receiving the pacs.002 message from the beneficiary bank and the CSM sending the pacs.008 message to the beneficiary bank;
- δ_3 approximates Phase C duration, computed as the time difference between the CSM sending the pacs.002 messages to both the originator and beneficiary banks and the CSM receiving the pacs.002 message from the beneficiary.

These processing times provide valuable operational insights into system behavior. The sum $\sum_{i=1}^{3} \delta_i$ estimates a lower bound for total transaction processing time, while $\sum_{i \in \{1,3\}} \delta_i$ represents time spent across CSM components. For instance, unexpected increases in processing times δ_1 or δ_3 may indicate failures in internal CSM infrastructure components, while anomalies in δ_2 processing times could suggest external failures involving participants or network connectivity problems.

9.3.2.2 *Time Series Construction through Data Aggregation*

Since instant payments occur at irregular intervals, individual transaction data are transformed into a regular time series including both processing time information and settled payment volume.

A time series of n settled transactions is defined as:

$$S = \{(t_i, \mathbf{p}_i)\}_{i=1}^{n} = \{(t_i, \delta_{1i}, \delta_{2i}, \delta_{3i})\}_{i=1}^{n}, \tag{9.2}$$

where the i-th observation $(t_i, \mathbf{p}_i)$ represents the instant payment $\mathbf{p}_i$ settled at time t_i.

To create a regular time series representation, S is resampled into m bins using a fixed sampling rate η, where the number of bins is determined by $m = \lceil (\omega - \alpha)/\eta \rceil$ with $\alpha \leq t_1$ as the start timestamp and $\omega \geq t_n$ as the end timestamp. The resampled time series is defined as:

$$S^{\eta} = \left\{(\tau_k, \tilde{\mathbf{p}}_k, v_k)\right\}_{k=1}^{m} \tag{9.3}$$

Within this resampled series, each bin b_k covers the time interval $\left[\alpha + (k-1)\eta,\ \alpha + k\eta\right)$, and the corresponding aggregated observation is defined by:

- $\tau_k = \alpha + (k-1)\eta$ is the representative timestamp for the k-th bin;
- $\tilde{\mathbf{p}}_k = (\tilde{\delta}_1, \tilde{\delta}_2, \tilde{\delta}_3)$ is the aggregated processing times for instant payments within bin b_k, computed using either the mean or median;
- $v_k = |b_k|$ represents the volume of settled payments within bin b_k.

Hence, each observation $\mathbf{s}_k = (\tilde{\mathbf{p}}_k, v_k) \in \mathbb{R}^4$ of the time series S^{η} captures aggregated processing times and payment volume for the k-th time bin.

9.3.3 *Explainable Anomaly Detection*

The anomaly detection framework consists of two complementary components addressing the subproblems defined in Section 9.3.1: a

detector identifying potential system failures by assigning anomaly scores to aggregated observations, and an explainer characterizing detected anomalies by determining severity and localization, enabling business impact assessment.

9.3.3.1 *Anomaly Detector*

The anomaly detector produces normalized anomaly scores in $[0, 1]$ for each feature. The detector $\hat{f}$ evaluates each new observation $\mathbf{s}_{m+1}$ from the data stream and produces an anomaly score vector $\boldsymbol{a}_{m+1} \in [0, 1]^4$:

$$\boldsymbol{a}_{m+1} = \hat{f}(\mathbf{s}_{m+1}), \tag{9.4}$$

with values closer to 1 indicating greater deviation from normal behavior. Subsequently, a binary anomaly label vector $\boldsymbol{y}_{m+1} \in \{0, 1\}^4$ is derived from the score vector:

$$\boldsymbol{y}^j_{m+1} = \mathbf{1}[\boldsymbol{a}^j_{m+1} > \theta^j] \tag{9.5}$$

where θ^j is a threshold parameter for feature j and $\mathbf{1}[\cdot]$ is the indicator function. Each component $\boldsymbol{y}^j_{m+1} = 1$ indicates that feature j is anomalous in the new observation. Both $\boldsymbol{a}_{m+1}$ and $\boldsymbol{y}_{m+1}$ serve as inputs to the explainer component.

9.3.3.2 *Anomaly Explainer*

The semantic meaning of the proposed features enables the interpretation of anomaly detector outputs through failure localization and incident severity classification, facilitating business impact assessment.

Failure Localization

The explainer leverages feature mapping to distributed system components to identify failure origins. It takes the binary anomaly label vector $\boldsymbol{y}_{m+1}$ and applies predefined localization rules $\mathcal{R}_{\mathcal{L}}$ to determine where in the system architecture the failure likely originated:

$$\hat{g}_{\mathcal{L}}(\boldsymbol{y}_{m+1}) \to \mathcal{L}. \tag{9.6}$$

The feature space inherently supports localization because different features naturally map to different system components. For instance, anomalies in $\tilde{\delta}_1$ or $\tilde{\delta}_3$ suggest internal CSM failures (A_1), while anomalies in $\tilde{\delta}_2$ indicate external failures (A_2).

Incident Severity Classification

The explainer also provides incident severity classification by analyzing anomaly score combinations across features. The classification function is represented as:

$$\hat{g}_{\mathcal{S}}(\boldsymbol{a}_{m+1}, \boldsymbol{y}_{m+1}) \to \mathcal{S}, \tag{9.7}$$

where the output is a severity level from set $\mathcal{S}$, based on domain knowledge $\mathcal{R}_{\mathcal{S}}$ about the relationship between feature deviation patterns and incident severity.

Business Impact Assessment

In distributed payment systems, business impact can vary widely even when technical severity appears similar, due to contextual factors such as timing (e.g., peak versus off-hours) and duration (brief spikes versus prolonged disturbances).

By providing both failure localization $\mathcal{L}$ and incident severity classification $\mathcal{S}$, the framework allows operational teams to assess business consequences and determine appropriate response strategies, prioritizing responses based on business impact rather than technical metrics alone.

For instance, an incident classified as major severity with internal localization (A_1) requires direct intervention, while the same severity incident with external localization (A_2) necessitates coordination with external service providers.

9.4 EXPERIMENTAL SETUP

We evaluated the proposed failure detection framework on TIPS [11, 56, 335] to assess its ability to (1) detect system failures, (2) localize failures, and (3) classify incident severity. The evaluation includes both

production data from a NSP incident and controlled experiments with systematic anomaly injection in a TIPS testing environment.

9.4.1 *Real-Time Feature Computation Architecture*

Real-time feature computation requires dedicated infrastructure to process payment message traces as they occur. As illustrated in Figure 9.2, an event-driven architecture is implemented comprising multiple distributed components for data collection, transformation, aggregation, and storage.

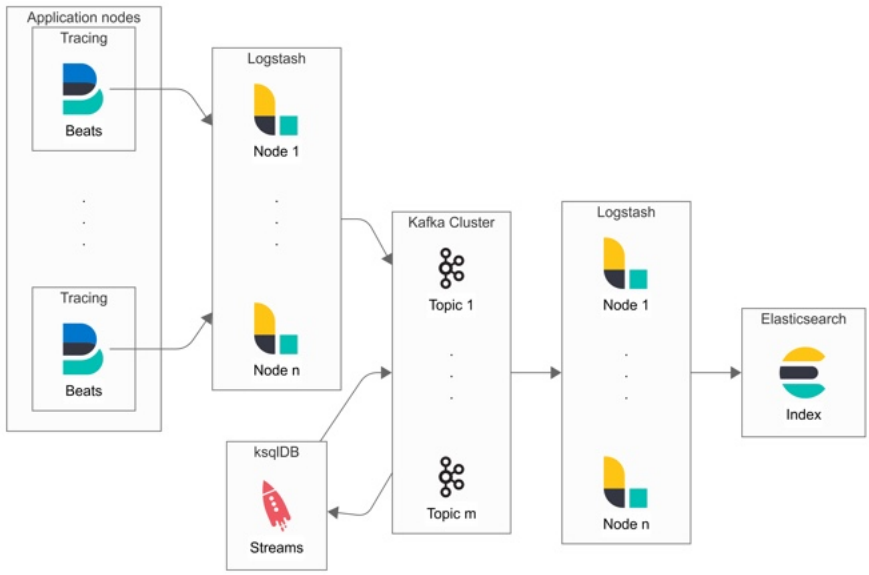


Figure 9.2: Real-time feature extraction architecture. Application nodes generate message traces, which are collected by Beats agents and processed by Logstash nodes. Messages flow through Kafka for buffering, where ksqlDB correlates messages to compute processing times. The processed features are then indexed in Elasticsearch for both storage and real-time analysis.

The extraction begins at application nodes within the TIPS infrastructure, where ISO 20022 payment message traces are captured by Elastic Beats agents and forwarded to a Logstash cluster for initial preprocessing. An Apache Kafka cluster provides message buffering

to handle variable transaction volumes without data loss. The core feature computation occurs in ksqlDB, which performs real-time stream processing to calculate processing times (δ_1, δ_2, δ_3) by matching corresponding messages and computing their time differences.

Processed features are indexed in Elasticsearch in two forms: raw data for historical analysis and model training, and aggregated metrics (with frequency η) for real-time anomaly detection. The aggregation process incorporates transaction volume feature (v), completing the multivariate time series representation of system behavior. This distributed architecture supports horizontal scaling of each component based on workload requirements, ensuring the feature extraction process maintains performance even during peak transaction periods.

9.4.2 *Experimental Testbed Design*

To systematically evaluate the failure detection framework across different scenarios, an experimental testbed was developed that generates realistic instant payment traffic with controlled anomaly injection capabilities. As shown in Figure 9.3, the testbed consists of two external simulators representing originator and beneficiary banks connected to a test instance of the TIPS platform mirroring the production environment.

These simulators, positioned in a network external to TIPS infrastructure (TIPSnet) to realistically model latency introduced by NSPs, exchange payment messages following processing flows identical to real transactions. The simulator is configured with message probability distributions matching production traffic patterns, implementing a multi-threaded application emulating multiple originators continuously generating payment streams at realistic volumes.

During test execution, controlled disturbances are injected into key TIPS components shown in Figure 9.3, including message router, message broker, and settlement core. This testbed enables simulation of various anomaly types, from processing slowdowns and network issues to simultaneous multi-component failures.

To ensure reproducibility and tracking, both traffic generation and anomaly injection are orchestrated through automated Ansible

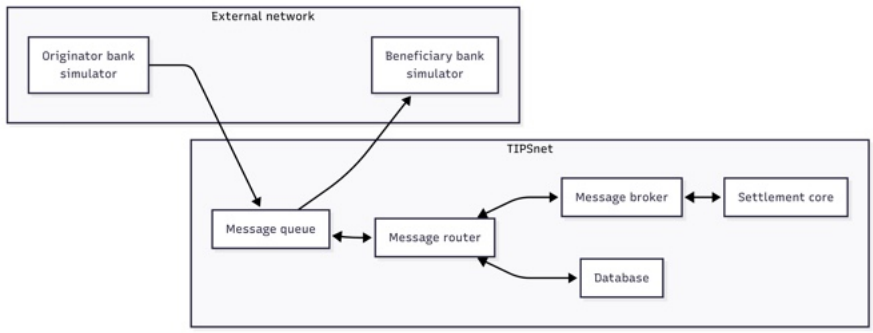


Figure 9.3: Architecture of the instant payment experimental testbed. External simulators representing the originator and beneficiary banks connect to the TIPS testing environment. Payment messages flow through the message queue and router to the core components. Anomalies can be injected at various points in the processing flow.

playbooks. Message traces from these simulations are processed through the feature pipeline in Figure 9.2, creating datasets stored in Elasticsearch indices.

9.4.3 *Datasets*

The architecture presented in Section 9.4.1 enabled the construction of two datasets: one containing production traces during a NSP incident, and another containing controlled experiments from the testbed with systematic anomaly injection.

The first dataset consists of transaction traces collected during August 2023, including a major NSP incident on 11 August impacting most TIPS participants for $\approx$ 15 minutes, causing processing slowdowns and payment timeouts. This production incident provides an authentic validation case under genuine operational conditions.

The second dataset derives from controlled experiments on the internal testbed, reproducing anomalous scenarios rare or difficult to isolate in production. This dataset contains $\approx$ 242,000 observations sampled at 15-second intervals ($\approx$ 42 days), of which the test set spans 19 days and comprises four testbed scenarios:

T1. Mild internal stress (≈ 7 min). Stress applied to a core component of TIPS to evaluate sensitivity to performance degradation.

T2. Stress on multiple internal components (≈ 5 min). Several TIPS core components stressed in parallel with moderate load.

T3. External participant disturbances (≈ 16 min). Disturbances impacting transactions from multiple participants outside TIPS simulated to assess discrimination between internal and external failure origins.

T4. Heavy internal stress (≈ 21 min). Multiple TIPS core components subjected to severe degradation, causing significant impact on transaction processing.

9.4.4 *Evaluation Metrics*

For evaluation, we employed both threshold-independent metrics (i.e., AUC-PR, AUC-ROC, PATE, VUS-PR, and VUS-ROC) and threshold-dependent metrics (i.e., MCC, range-based metrics, and affiliation metrics). We also computed operational metrics such as FAR and EDR, and for detected events only, MTTD. Metric definitions are in Section 2.5.2.

Additionally, to assess agreement between model anomaly characterization and ground truth annotations by domain experts, we employed Cohen's kappa coefficient [82]. This metric quantifies concordance between two sets of labels beyond chance. The coefficient is calculated as:

$$\kappa = \frac{p_o - p_e}{1 - p_e} \tag{9.8}$$

where p_o represents observed agreement between model outputs and ground truth labels, while p_e denotes expected agreement under random classification. Chance agreement p_e is computed based on marginal distributions of both predicted and true labels [12].

9.4.5 *Implementation Details*

For the anomaly detector component, we compare an instance of the MoGE-cVAE architecture (presented in Section 8.3) with the anomaly detection capabilities of Elastic Machine Learning (EML) [104]. Elastic's anomaly detection uses a hybrid ensemble learning approach combining clustering, time series decomposition, Bayesian distribution modeling, and correlation analysis [104]. The engine establishes a probabilistic baseline from historical data while dynamically adapting to new observations, computing deviation scores for each time bucket with multi-stage noise reduction and statistical significance ranking. We utilize the same explainer component for anomaly characterization.

9.4.5.1 *Anomaly Detector Component*

We trained MoGE-cVAE experts, where conditional VAEs have an encoder with two layers [256, 128] and a decoder with layers [128, 256]. Training uses MSE as reconstruction loss and KL penalty with annealing. Optimization is performed with Adam optimizer, batch size 32, and KL warmup policy. Performance is evaluated using MAE and Wasserstein distance.

For hyperparameter selection with a compute budget of 15 trials, we used randomized grid search over 64 candidate hyperparameter combinations. The search space included KL annealing (maximum factor: 0.3–0.5), Adam learning rate $(1 \cdot 10^{-4}, 5 \cdot 10^{-5})$, latent dimensions $\{3, 6\}$, and free-bits $\{0.0, 0.5\}$. Experts were ranked using MAE and Wasserstein distance. For deployment, we selected one expert per category based on this ranking. Table 9.1 shows selected experts, where epoch denotes early-stop epoch within its corresponding run.

We compared MoGE-cVAE with two Elastic machine learning configurations. The first configuration is a multi-metric model (EML-M) jointly analyzing the four features $(\tilde{\delta}_1, \tilde{\delta}_2, \tilde{\delta}_3, v)$ without influencers, using `low_sum` detector for v and `high_mean` for $\tilde{\delta}_1, \tilde{\delta}_2, \tilde{\delta}_3$. The second is a partitioned multi-metric model (EML-MP), introducing day-of-week as both `partition_field` and `influencer`, creating a sub-model

Category	LD	LR	KLF	FB	KWS	KWE	Epoch
0	3	0.0001	0.3	0.0	10	80	30
1	3	0.0001	0.5	0.0	0	80	16
2	3	0.0001	0.3	0.5	10	80	16
3	3	0.0001	0.3	0.5	10	80	31
4	3	0.0001	0.3	0.5	10	40	49
5	3	0.0001	0.3	0.5	10	80	15
6	3	0.0001	0.3	0.5	10	80	39

Abbreviations: LD: latent dimension, LR: learning rate, KLF: maximum KL factor, FB: free bits, KWS: KL warmup start epochs, KWE: KL warmup duration epochs.

Table 9.1: Selected expert hyperparameters per category. The epoch indicates the early-stop epoch within the selected run.

for each day. Both variants adopt a model memory limit of 500 MB and a model pruning window of 30 days.

Experiments were conducted in an unsupervised learning setting using aggregation frequency $\eta = 15$ seconds. For Elastic models, two threshold configurations were evaluated based on risk indicators: a high threshold corresponding to scores exceeding 70, and a critical threshold for scores exceeding 90. For MoGE-cVAE, custom thresholds were employed based on semantically meaningful values specific to each feature: a reduction of more than 100 payments compared to expected values for payment count feature v, latency exceeding 10ms for processing time features $\tilde{\delta}_1$ and $\tilde{\delta}_3$, and latency exceeding 1s for queue time feature $\tilde{\delta}_2$.

9.4.5.2 *Anomaly Explainer Component*

To instantiate the explainer component for experimental validation, two simple rulesets $\mathcal{R}_\mathcal{L}$ and $\mathcal{R}_\mathcal{S}$ are defined based on TIPS architectural knowledge. Localization ruleset $\mathcal{R}_\mathcal{L} = \{L1, L2\}$ distinguishes between internal and external failure origins by analyzing patterns in binary anomaly label vector:

L1. *Internal*. When anomalies occur in processing times $\tilde{\delta}_1$ or $\tilde{\delta}_3$ without corresponding anomalies in $\tilde{\delta}_2$, this pattern suggests

that TIPS internal components experience delays while external communication remains unaffected, indicating an internal failure origin.

L2. *External*. When anomalies appear exclusively in processing time $\tilde{\delta}_2$ while $\tilde{\delta}_1$ and $\tilde{\delta}_3$ remain normal, this indicates TIPS processes messages normally but external entities exhibit delayed responses, suggesting an external failure origin.

Incident severity classification ruleset $\mathcal{R}_S = \{S1, S2\}$ maps anomaly patterns to incident severity levels:

S1. *Minor*. Anomalies in processing times ($\tilde{\delta}_1$, $\tilde{\delta}_2$, or $\tilde{\delta}_3$) without corresponding volume drops indicate performance degradation. We classify these as minor incidents.

S2. *Major*. Anomalies impacting settled payment volume v indicate major incidents.

For practical implementation, we synthesize rulesets $\mathcal{R}_L$ and $\mathcal{R}_S$ using masks on binary label vector $\boldsymbol{y}_k \in \{0,1\}^4$ for each observation produced by the anomaly detector. To capture operational significance of specific anomaly patterns, we encode the binary label vector into a single integer code preserving both anomaly pattern and relative severity.

We define an anomaly code using weighted binary encoding:

$$c_k = \sum_{j=1}^{4} w_j \cdot \boldsymbol{y}_k^j, \tag{9.9}$$

where w_j are predefined weights assigned to each feature. By carefully selecting these weights, we ensure codes naturally reflect operational severity ordering. For instance, using weights $\mathbf{w} = [8, 4, 2, 1]$ for features $[v, \tilde{\delta}_1, \tilde{\delta}_2, \tilde{\delta}_3]$ respectively, we obtain the mapping shown in Table 9.2.

This encoding scheme provides several advantages for facilitating operator interpretation and incident categorization: it uniquely identifies each anomaly pattern, preserves severity ordering through weight selection, and enables efficient pattern matching for explainability rules.

y_k^v	$y_k^{\tilde{\delta}_1}$	$y_k^{\tilde{\delta}_2}$	$y_k^{\tilde{\delta}_3}$	Code	$\mathcal{L}$	$\mathcal{S}$	BI
0	0	0	0	0	None	None	No
0	0	1	0	2	External	Minor	No
0	1	0	1	5	Internal	Minor	No
1	0	1	0	10	External	Major	Yes
1	1	0	1	13	Internal	Major	Yes

Table 9.2: Anomaly encoding scheme mapping binary label vectors to severity classes. Weights $[8, 4, 2, 1]$ ensure codes reflect increasing severity while preserving pattern distinction, with semantic meaning correlated to business impact (BI). Each row shows whether an anomaly is detected (1) or not (0) for each feature.

9.5 RESULTS

This section presents experimental results obtained from both testbed scenarios and the NSP case study. Controlled testbed experiments enabled systematic evaluation across failure conditions challenging to reproduce in production. The NSP incident validates the framework's capability to distinguish between internal and external failure origins in a real-world scenario, where post-incident root cause analysis confirmed that network connectivity issues at the NSP level disrupted communication between TIPS and multiple participants.

9.5.1 *Testbed Scenarios*

The testbed evaluation assessed detection performance across four controlled failure scenarios (T1–T4). We first report threshold-independent metrics to evaluate the models' ability to separate anomalous from normal observations, followed by threshold-dependent metrics and anomaly characterization results.

Table 9.3 presents threshold-independent metrics averaged across all testbed scenarios and features. These metrics assess separability between normal and anomalous patterns without requiring threshold selection.

Metric	EML-M	EML-MP	MoGE-cVAE
AUC-PR	0.7947	0.7649	**0.9907**
AUC-ROC	0.9911	0.9410	**1.0000**
PATE	0.7977	0.7643	**0.9917**
VUS-PR	0.2978	0.2283	**0.3242**
VUS-ROC	0.7059	0.6844	**0.8414**

Table 9.3: Threshold-independent metrics averaged across all testbed scenarios and features. These metrics evaluate anomaly separability without requiring threshold selection.

Table 9.4 reports threshold-dependent metrics averaged across all testbed scenarios and features. These metrics evaluate detection performance at specific operating points determined by threshold configurations described in Section 9.4.5.1.

Figure 9.4 illustrates anomaly characterization results for scenarios T3 and T4, showing how different models classify observations according to the encoding scheme defined in Table 9.2. The visualization enables comparison of model outputs against ground truth annotations established by domain experts.

Table 9.5 quantifies the agreement between model-predicted anomaly characterizations and ground truth annotations using Cohen's kappa coefficient (Eq. 9.8). This metric measures concordance beyond chance agreement, providing insight into how well each model's explanations align with expert-annotated failure localization and severity classifications.

9.5.2 *Case Study: Network Service Provider Incident*

The NSP incident evaluation assesses framework performance on a real production incident with publicly known ground truth. This incident, occurring in August 2023, affected connectivity between TIPS and multiple participants for $\approx$ 15 minutes, providing an authentic validation case under genuine operational conditions.

Table 9.6 presents threshold-independent metrics for this case study. Both models achieve strong separability between normal and anoma-

Metric	EML-M (Critical)	EML-M (High)	EML-MP (Critical)	EML-MP (High)	MoGE-cVAE (Custom)
MCC	0.6603	0.6569	0.6516	0.4057	**0.9446**
Precision_T	0.4913	0.4613	0.4998	0.0958	**0.8201**
Recall_T	0.6179	0.7133	0.7412	0.7500	**0.8492**
F_{1_T}	0.5442	0.5547	0.5612	0.1635	**0.7662**
$\text{P}_{\text{precision}}$	0.8703	0.7672	0.9031	0.5968	**0.9822**
P_{recall}	0.6250	0.8302	0.7500	0.7686	**1.0000**
P_{F_1}	0.8500	0.7925	0.9435	0.6437	**0.9908**
FAR/h	0.0401	0.0984	0.0626	0.6450	**0.0049**
EDR	0.6250	0.8125	0.7500	0.7500	**1.0000**
MTTD^*_{obs}	0.6667	17.2500	0.2778	**0.0000**	0.3750

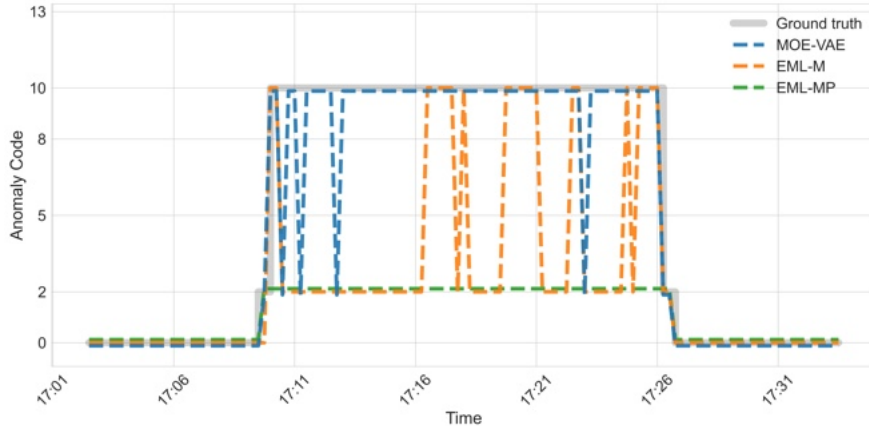
* MTTD is computed only over detected events. EDR indicates the fraction of anomaly events detected. When EDR = 0, MTTD is undefined.

Table 9.4: Threshold-dependent metrics (point-based, range-based, affiliation-based, and operational) averaged across all testbed scenarios and features. Each column corresponds to a specific (model, threshold) configuration.

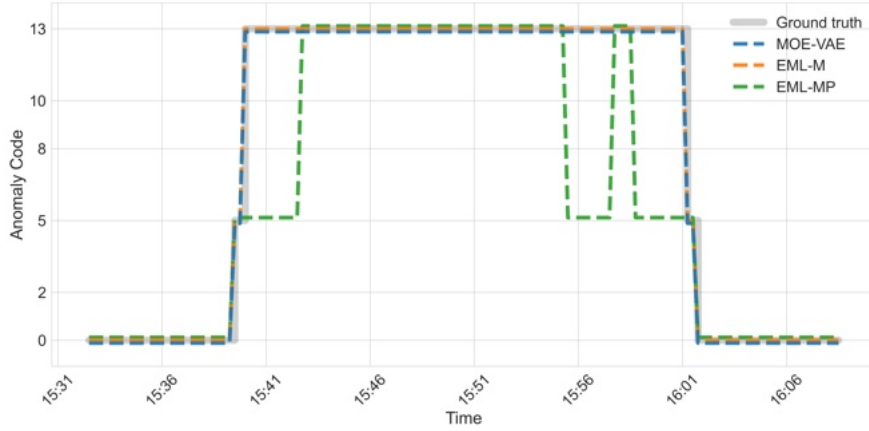
Model	Threshold		
	High	Critical	Custom
EML-M	0.393	0.409	N/A
EML-MP	0.166	0.396	N/A
MoGE-cVAE	N/A	N/A	**0.837**

N/A: not applicable.

Table 9.5: Cohen's kappa coefficient measuring agreement between predicted anomaly characterizations and ground truth annotations across models and thresholds. Higher values indicate stronger concordance with expert-annotated failure localization and severity classifications.



(a) Scenario T3: MoGE-cVAE and EML-M show agreement in classifying the external major event (code 10), while EML-MP consistently underestimates severity, classifying it only as external minor (code 2).



(b) Scenario T4: MoGE-cVAE and EML-M demonstrate accurate agreement with ground truth for the internal major event (code 13), while EML-MP exhibits delayed classification of the same event.

Figure 9.4: Anomaly characterization comparison for selected testbed scenarios using encoding weights $\mathbf{w} = [8, 4, 2, 1]$ for features $[v, \tilde{\delta}_1, \tilde{\delta}_2, \tilde{\delta}_3]$. Gray trace shows ground truth, dashed blue represents MoGE-cVAE classifications, dashed orange shows EML-M classifications (critical threshold), and dashed green indicates EML-MP classifications (critical threshold).

Metric	EML-M	MoGE-cVAE
AUC-PR	**0.9294**	0.9182
AUC-ROC	**0.9996**	0.9670
PATE	**0.9507**	0.9265
VUS-PR	**0.7267**	0.4424
VUS-ROC	0.7171	**0.7520**

Table 9.6: Threshold-independent metrics for the NSP incident case study, averaged across features.

lous observations, with EML-M showing slightly higher performance on most metrics.

Table 9.7 reports threshold-dependent metrics at specified operating points. Results reveal notable differences in operational performance: MoGE-cVAE achieves lowest detection latency (0.5 observations) and produces no false alarms, while EML-M with high threshold detects all events but with longer delays (12 observations) and higher false alarm rate.

Figure 9.5 illustrates anomaly characterization comparison, showing how models classify observations according to failure localization and severity encoding scheme. MoGE-cVAE maintains closer temporal alignment with ground truth throughout incident duration, whereas EML-M exhibits delayed initial detection and generates additional noise during recovery phase.

Agreement between model predictions and ground truth, quantified by Cohen's kappa coefficient, confirms these observations: EML-M with critical threshold achieved $\kappa = 0.198$, EML-M with high threshold achieved $\kappa = 0.724$, and MoGE-cVAE with custom threshold achieved $\kappa = 0.839$. These values indicate MoGE-cVAE provides anomaly characterizations more consistent with expert annotations.

Metric	EML-M (Critical)	EML-M (High)	MoGE-cVAE (Custom)
MCC	0.2347	0.8026	**0.9120**
Precision_T	0.3333	0.8833	**1.0000**
Recall_T	0.0108	0.3522	**0.5100**
F_{1_T}	0.0210	0.4426	**0.5429**
$\text{P}_{\text{precision}}$	0.9978	0.9994	**1.0000**
P_{recall}	0.4975	0.9994	**1.0000**
P_{F_1}	0.9964	0.9994	**1.0000**
EDR	0.5000	**1.0000**	**1.0000**
FAR/h	0.0811	0.0811	**0.0000**
MTTD^*_{obs}	36.0000	12.0000	**0.5000**

* MTTD is computed only over detected events. EDR indicates the fraction of anomaly events detected. When EDR = 0, MTTD is undefined.

Table 9.7: Threshold-dependent metrics for the NSP incident, averaged across features and evaluated at specified threshold configurations. Each column corresponds to a specific (model, threshold).

9.6 DISCUSSION

The experimental results validate the efficacy of the feature engineering approach for failure detection in instant payment infrastructures. This section discusses the effectiveness of the anomaly detection framework, its comparative advantages over traditional monitoring approaches, and the explainability aspects of the proposed solution.

9.6.1 *Effectiveness of the System State Representation*

Processing times between consecutive ISO 20022 message exchanges can be effectively engineered into meaningful features through a systematic approach that considers message volumes and processing durations. By treating each transaction as a sequence of standardized messages, temporal intervals between consecutive messages reveal

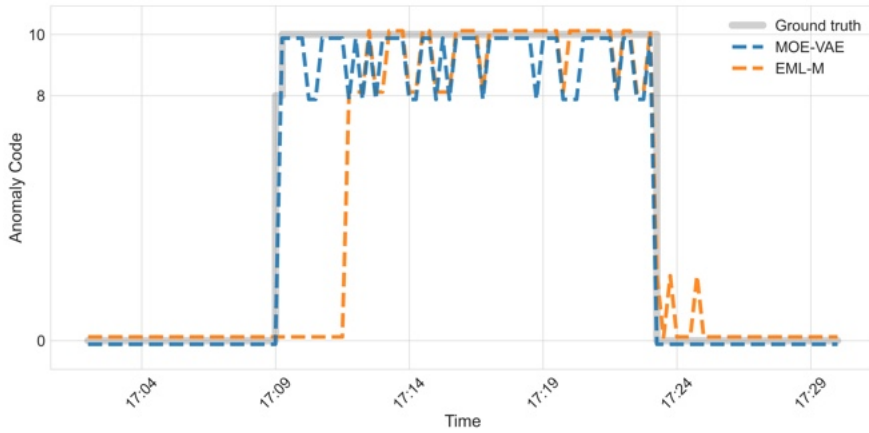


Figure 9.5: Anomaly characterization for the NSP incident case study using encoding weights $\mathbf{w} = [8, 4, 2, 1]$ for features $[v, \tilde{\delta}_1, \tilde{\delta}_2, \tilde{\delta}_3]$. Gray trace shows ground truth, dashed blue represents MoGE-cVAE classifications, and dashed orange shows EML-M classifications (high threshold). MoGE-cVAE demonstrates stronger agreement with ground truth, while EML-M exhibits delayed detection and increased noise in later phases.

operational efficiency of different system components and process phases.

When aggregated across multiple transactions, these processing times create a compact representation of the payment system's holistic state. Unlike traditional infrastructure metrics operating in isolated data silos, this approach captures end-to-end transaction flows spanning component boundaries. Features inherently map to different process phases executed on different system components, enabling a semantic layer bridging technical infrastructure metrics with business process visibility.

The value of this feature engineering approach is demonstrated through its ability to detect anomalies corresponding to observed disturbances across various failure modes, validated in both real-world incidents and controlled simulations. Across all testbed scenarios, MoGE-cVAE achieved high anomaly separability (see Table 9.3), outperforming Elastic models. Similarly, for the NSP incident, models demonstrated strong separability performance (see Table 9.6). Fea-

tures successfully capture irregular behaviors in payment systems by identifying increases in processing times or reductions in settled payment volumes, which serve as indicators of potential system failures.

9.6.2 *Problem Formulation and Detection Performance*

In SCT Inst, significant observability challenges arise from distributed systems spanning multiple transaction participants, where no single actor has complete transaction visibility. The CSM perspective is particularly valuable because it provides a centralized vantage point to observe interactions between multiple distributed participants in the payment ecosystem. This positioning enables detecting failures manifesting at system integration level, which might be invisible to individual components operating normally.

By analyzing patterns across multiple engineered features simultaneously, the system can identify subtle changes in operational behavior preceding failures. MoGE-cVAE demonstrates strong operational performance across both testbed scenarios and the NSP incident, compared to detection rates of 62.5–81.25% for Elastic models in testbed scenarios. Furthermore, MoGE-cVAE achieved the lowest FAR of 0.0049 false alarms per hour in testbed scenarios and zero false alarms in the NSP incident, significantly outperforming Elastic configurations. This low false alarm rate minimizes alert fatigue for operators monitoring infrastructure.

9.6.3 *Advantages over Traditional Monitoring*

Traditional monitoring excels at resource utilization tracking and assessment of individual system component health and performance but faces fundamental limitations when dealing with incidents involving external elements or system integration issues. The proposed approach complements traditional monitoring systems by providing a business-oriented semantic layer bridging technical infrastructure metrics and operational impact assessment. Appendix b shows how

the proposed framework integrates with existing monitoring infrastructures.

The NSP incident illustrates these limitations: such incidents manifest as business-level anomalies (reduced settled payments) without corresponding infrastructure alerts, where component-level monitoring shows normal operation while integrated service experiences significant degradation. The proposed framework addresses this observability gap by leveraging features spanning component boundaries and capturing end-to-end transaction flows. In this real-world scenario, MoGE-cVAE demonstrates its ability to accurately detect and temporally align with ground truth events.

The framework's computational efficiency represents another significant advantage. Unlike AIOps solutions processing thousands of real-time metrics, the proposed feature engineering focuses on four compact features representing business-relevant system states, making it computationally lightweight while maintaining high semantic value. This efficiency is achieved without sacrificing detection quality, as evidenced by performance metrics across all evaluation dimensions.

9.6.4 *Explainability and Anomaly Characterization*

Interpretation of detected anomalies is achieved through an explainability-by-design approach, contrasting with post-hoc explanation methods attempting to explain opaque model decisions after detection. The key to this interpretability lies in the semantic richness of engineered features, inherently mapping to different process phases executed on different system components. Simplified modeling of transactions facilitates straightforward interpretation of results by IT operators.

The framework enables failure localization by analyzing combinations of detected anomalies to identify whether incident origins are internal or external to the payment system. This capability is fundamental to effective incident response, immediately guiding operators toward appropriate system components or external dependencies requiring attention. The natural mapping between features and system

components eliminates the need for complex interpretation algorithms or specialized domain knowledge to understand problem locations. Different combinations of anomalies within multivariate time series features provide initial discrimination to understand where the problem is occurring and guide IT operators toward the root cause.

Quantitative assessment of anomaly characterization quality through Cohen's kappa coefficient demonstrates the framework's capability in providing actionable explanations. MoGE-cVAE achieved substantial agreement with expert annotations ($\kappa = 0.837$ for testbed scenarios and $\kappa = 0.839$ for NSP incident), compared to Elastic models (best performance: $\kappa = 0.409$ for testbed scenarios and $\kappa = 0.724$ for NSP incident). These results indicate that framework explanations align closely with domain expert interpretations of failure localization and severity.

Incident severity classification is achieved through domain knowledge integration enabling business impact assessment. The framework classifies incident severity based on actual business impact rather than technical severity alone, providing clear explanations guiding operator responses. As illustrated in Figure 9.4, MoGE-cVAE correctly distinguished between external minor events (code 2) and external major events (code 10) in scenario T_3, while also accurately classifying internal major events (code 13) in scenario T_4. This capability enables more nuanced and appropriate response strategies compared to binary anomaly detection approaches, recognizing that anomalies do not always signify incidents with business impact.

Accurate forecasting enables thresholds with clear operational meaning. For instance, thresholds can be defined based on interpretable criteria such as volume falling below a specific level or latency exceeding defined values (e.g., 10 ms or 1 second). Combined with the proposed problem formulation, this yields outputs that are clear, simple, and interpretable for operators.

The explainability of this framework accelerates operators' daily investigations by providing immediate access to reconstructed transaction flows, eliminating the need for laborious manual correlation of messages across distributed system components. This reduction in

cognitive load enables effective incident response without requiring deep technical expertise about underlying system architecture.

9.6.5 *Operational Advantages*

Once deployed, the model operates autonomously by inferring day type labels (e.g., weekday) from timestamps to condition the latent vector, requiring no explicit user input. This enables single-shot estimation of observation distributions for future days, allowing real-time comparison with streaming data for immediate outlier detection. The model can be continuously retrained on new data to adapt to emerging trends.

Independence from recent observations offers practical operational advantages. Unlike sequential models requiring continuous historical windows, the pointwise generative approach remains functional even when recent data are unavailable or incomplete. This characteristic reduces prediction lag by generating forecasts directly without requiring data accumulation.

The MLP-based architecture enables efficient training and inference. This simplicity yields advantages in model management, maintenance, resource allocation, and energy consumption. The capacity for long-term predictions further enhances these benefits by reducing computational requirements compared to models continuously processing incoming data streams.

9.6.6 *Limitations and Future Directions*

The current approach cannot anticipate failures occurring without preceding warning signs, such as complete network disconnections immediately terminating all connections. Such events provide no gradual performance degradation for advance detection, highlighting the need for complementary monitoring mechanisms operating at the infrastructure layer.

The approach does not directly identify root causes but, by enabling failure localization, reduces time required to initiate root cause analysis (RCA), which aims to map an incident to its underlying fault.

Future research could explore combining insights from the proposed approach with traditional infrastructure monitoring data to facilitate comprehensive RCA. By correlating business-level anomalies with low-level system metrics, this integration may enable automated remediation strategies responding to detected incidents without manual intervention.

While implementation focused on the CSM perspective, the underlying anomaly detection framework demonstrates considerable flexibility and may be adapted for other SCT Inst process actors, such as individual PSPs. This adaptation would create opportunities for multi-level monitoring across different ecosystem participants, potentially enabling more precise failure localization when incidents originate outside CSM infrastructure.

Another promising extension involves integrating the approach with LLMs to enhance operational usability through natural language interfaces. Such integration could provide operators with enriched explanations incorporating additional contextual information and recommended corrective actions in natural language, reducing technical expertise required for effective incident response. This direction aligns with emerging trends in AIOps emphasizing human-AI collaboration for operational intelligence.

Part V

CONCLUSIONS AND PERSPECTIVES

Beyond forecasting and detection.

10 CONCLUSIONS

This dissertation examines the potential of generative deep learning for seasonal time series synthesis via the GenTT framework, focusing on forecasting and anomaly detection. We propose an alternative to sequential processing based on sliding windows, leveraging explicit geometric time encodings to enable history-free conditional generation.

While sliding window methods are effective in many scenarios, they face practical limitations, including sensitivity to hyperparameter tuning and error accumulation in long-horizon forecasting. These constraints are critical in high-frequency multivariate time series, where real-time processing requires lightweight and scalable solutions.

The research pursued three main objectives: designing a mechanism to enable generative models to synthesize seasonal time series without sequential preprocessing; evaluating this approach against state-of-the-art methods; and validating the framework in real-world high-frequency instant payment scenarios under strict operational constraints. By formalizing a geometric representation of time, we developed a framework capable of non-sequential generation, which enabled the construction of real-time, explainable anomaly detection systems that bridge technical metrics with business impact assessment.

10.1 SUMMARY OF MAIN FINDINGS

We demonstrated that accurate synthesis of seasonal time series through the proposed GenTT framework offers a viable and often superior alternative to traditional sequential methods, particularly for high-frequency and noisy data. We instantiated the framework using two generative architectures: GAN-based models, which produced sharp, realistic predictions suitable for forecasting, and VAE-based

models, which provided robust baseline estimation advantageous for anomaly detection in high-noise environments.

This work addressed the following research questions:

- *Geometric representation of temporal information (RQ1).* We formalized HTE, an invertible encoding mapping timestamps to positions on a helix, where seasonal recurrence corresponds to angular proximity and temporal progression to axial displacement. This deterministic geometric structure ensures observations at similar seasonal phases remain proximal regardless of chronological distance, enabling the model to exploit seasonal patterns through spatial relationships.
- *Conditioning mechanisms for generative synthesis (RQ2).* We designed the GenTT framework, which leverages time encodings such as HTE to enable deep learning models to synthesize seasonal time series by conditioning on temporal representations. This introduces an alternative approach where time is treated as an explicit conditioning variable rather than an implicit ordering dimension, enabling independent observation generation and eliminating sliding window preprocessing during both training and inference.
- *Impact on deep learning architectures (RQ3a).* Empirical evaluation demonstrates that HTE consistently improves pattern recognition capabilities across MLP, CNN, and LSTM architectures when applied to seasonal time series forecasting. The consistency of these improvements indicates that benefits derive from the temporal representation itself rather than architecture-specific optimizations.
- *Long-term forecasting performance (RQ3b).* GenTT-based models achieved competitive or superior performance in long-term forecasting scenarios compared to established autoregressive approaches. This advantage stems from eliminating compounding errors: while autoregressive models propagate initial inaccuracies, our conditional generation approach predicts complete seasonal periods in a single inference step based solely on timestamp-derived conditions.

- *Anomaly detection performance (RQ4)*. The GenTT framework demonstrated substantial advantages over prediction-based sliding window approaches in both detection performance and resource efficiency. The framework achieved a 1,255-fold reduction in energy consumption compared to autoregressive LSTM methods while maintaining stable performance with up to 15% missing data and showing high resilience to noise. Training time scaled efficiently with dimensionality, and inference time remained independent of input dimensions, rendering the approach suitable for edge computing environments where minimizing continuous computational overhead is essential. Furthermore, the system is naturally immune to contamination from recent outliers.
- *Real-time streaming anomaly detection (RQ5)*. Validation on high-frequency instant payment systems confirmed the GenTT framework's practical viability in resource-constrained environments. The modular MoGE architecture enables selective model updates when concept drift occurs, reducing both training time and energy consumption compared to complete model retraining.
- *Failure localization and impact assessment (RQ6)*. Application to instant payment infrastructures demonstrated that the anomaly detection framework built on GenTT facilitates early detection of service degradation and failure localization across distributed components. Furthermore, its explainability provides real-time insights to guide remediation efforts and assess business impact.

10.2 CRITICAL DISCUSSION

Empirical evidence confirms that HTE combined with generative models effectively captures temporal dependencies through geometric constraints, guiding models toward temporally consistent outputs without requiring lookback windows. This suggests that the encoding successfully embeds the necessary temporal structure for accurate prediction.

The improved long-term performance of the GenTT-based model compared to autoregressive approaches stems from preventing error accumulation through independent conditional generation. While sliding window methods face a trade-off between window size and context retention, our approach leverages the entire historical dataset during training to learn global patterns. This framework eliminates the latency-accuracy trade-off inherent to sliding windows and removes the complexity of optimizing window size parameters.

Computational advantages are particularly pronounced in high-frequency streaming scenarios. Timestamp encoding enables both accurate baseline generation and zero-latency anomaly detection by decoupling contextual requirements from response time. By eliminating window-based processing, the hyperparameter search space is reduced, as window-related parameters such as lookback size, stride, and overlap are no longer present.

Furthermore, history-free inference provides operational resilience: predictions can be generated solely from timestamps without maintaining historical buffers or requiring ordered data streams. The lightweight MLP-based architecture enables deployment on consumer hardware without GPU acceleration, making advanced generative modeling accessible in edge environments.

10.3 LIMITATIONS

The primary limitation of the proposed method is its dependence on explicit seasonality. The approach requires a priori specification of the seasonal period, a domain knowledge requirement that traditional methods might discover automatically or implicitly. While this information is typically available in seasonal forecasting applications, the framework does not currently include automated methods for period determination. Consequently, the approach is less suitable for non-seasonal time series or data with highly irregular, non-cyclic patterns.

Although high-frequency time series often exhibit multiple complex seasonalities (e.g., daily, weekly, and annual patterns), the current implementation focuses on encoding a primary seasonal cycle. While the framework can theoretically accommodate multiple

periodicities through composite encodings, experiments primarily validated the method on series where a dominant seasonality could be explicitly identified.

Additionally, eliminating sliding windows, while advantageous for long-term stability and efficiency, precludes immediate adaptation to the most recent observations. This characteristic is beneficial in stable contexts where temporary fluctuations should not influence the baseline, but it disadvantages the system when immediate adaptation to rapid concept drift is required. In highly dynamic environments, hybrid mechanisms or periodic retraining strategies are necessary to address significant distributional shifts.

10.4 IMPLICATIONS AND IMPACT

The research presented in this thesis has direct implications for real-time infrastructure. The proposed methods are being adopted within TIPS, the instant payment settlement service of the Eurosystem. Driven by the Instant Payments Regulation [116], the onboarding of new currencies including the Swedish krona [112] and Danish krone [490], the extension to the Western Balkans [23], and cross-currency settlement capabilities [71, 336], the expansion of the TIPS scope makes robust real-time failure detection essential. In such time-critical environments where infrastructure issues immediately impact end users, rapid failure localization and business impact assessment are vital.

Beyond the financial domain, the GenTT framework holds broader interdisciplinary potential. The increasing prevalence of high-frequency sensors across sectors, including smart grids, intelligent transportation systems, healthcare monitoring, and cybersecurity, creates growing demand for lightweight, energy-efficient anomaly detection solutions. The proposed generative approach, capable of running on resource-constrained hardware with a lower carbon footprint than sliding window methods, offers a sustainable path for deploying advanced AI at the edge. Furthermore, GenTT could also be extended to creative domains dealing with periodic signals, such as audio generation.

11 FURTHER RESEARCH DIRECTIONS

The GenTT framework extends beyond seasonal time series forecasting and anomaly detection. This chapter outlines future research directions, covering theoretical and methodological enhancements as well as applications in cybersecurity, signal processing, and creative arts.

Generative Residual Decomposition for Non-Seasonal Dynamics

To extend GenTT to non-seasonal or weakly periodic data, a generative residual learning approach could leverage the components learned through encodings such as HTE, while delegating stochastic residuals to foundation models or time series-adapted LLMs. This approach would augment GenTT with the expressive capacity of pre-trained transformers for irregular dynamics. This would reframe synthesis as a neural signal decomposition task, extending applicability to mixed-dynamic domains such as biomedical signals and energy systems.

Furthermore, GenTT is suitable for uncertainty quantification. Since the inference process generates a set of samples per seasonal period, this empirical distribution can be directly leveraged to estimate aleatoric uncertainty. Instead of collapsing samples into single values per bucket, the framework can output empirical quantiles to define prediction intervals. This enables the generation of statistically calibrated confidence bands with guaranteed coverage probabilities without requiring architectural modifications.

Theoretical Formalization and Geometric Extensions

Although the geometric intuition behind HTE is empirically robust, formalizing these properties within manifold learning theory repre-

sents a promising direction. A formal proof that the helical embedding disentangles cyclic temporal dynamics in a structured geometric space would strengthen the theoretical foundation, facilitating stable conditional distributions for observations. Theoretical challenges include determining the expressiveness of the mapping (i.e., the class of seasonal functions it can faithfully represent) and establishing identifiability conditions that guarantee injectivity under irregular sampling or overlapping multi-scale periodicities.

Additionally, alternative geometric representations could replace HTE within the GenTT framework. For instance, toroidal encodings could capture multiple nested periodicities simultaneously, while polar coordinate systems might offer computational advantages. Exploring progression functions $p(t)$ inspired by delay embedding techniques from dynamical systems theory could further enhance the framework's ability to capture dynamics in complex time series.

Disentangled Latent Representations for Interpretability

Beyond the feature-based explainability presented in Chapter 9, intrinsic interpretability can be achieved through latent space disentanglement. This involves enforcing constraints via regularization techniques, such as β-VAE formulations or mutual information maximization, encouraging the model to map distinct dimensions of the latent vector $\mathbf{z}$ to semantically meaningful factors of variation.

A disentangled latent space would enable controlled interventions on the learned representation. By systematically perturbing individual latent dimensions, we could simulate counterfactual scenarios, such as investigating how an anomaly would manifest under different intensity levels. This would transform the generative model from an opaque synthesizer into an interpretable simulator, where explanations derive directly from the underlying generative factors.

Automated Hyperparameter Optimization

While the proposed framework eliminates sliding window hyperparameters, architectural parameters such as network depth and

latent space dimensionality still require optimization. Manual model selection is time-consuming and prone to bias. Integrating Neural Architecture Search (NAS) [107] to automate the search for optimal architectures would streamline model development.

Time Series Disaggregation

Primary applications of GANs in time series have focused on data augmentation or imputation [24, 125, 183]. However, GenTT could offer an approach to temporal disaggregation, i.e., increasing time series frequency while preserving temporal dynamics and integral constraints. By adjusting the sampling strategy to generate more points per seasonal period and defining finer temporal bins during aggregation, the framework could produce higher-frequency time series that respect the dynamics captured during training. Unlike naive upsampling methods that uniformly distribute aggregate values, this generative approach would yield temporally informed disaggregation that approximates original high-frequency patterns.

Low-and-Slow Attack Detection

The independence of GenTT from recent observations would be advantageous for detecting stealthy attacks that slowly degrade system performance. Traditional sliding window-based approaches often adapt to gradually increasing attack traffic, interpreting malicious patterns as legitimate trends (as shown in Figure 8.11, where models with smaller sliding windows closely track anomalies, exhibiting excessive adaptation to recent inputs). In contrast, the proposed method generates predictions based on patterns learned during training, remaining resistant to real-time manipulation.

Consider a slow-rate denial-of-service attack [299, 440] that increases resource consumption gradually to avoid threshold-based alarms. While a sliding window-based model might incorporate this increase into its baseline, the GenTT-based approach maintains predictions anchored to historical normal behavior. This would provide

early warning capabilities for low-and-slow attacks and quality-of-service degradation that often evade adaptive detection systems.

Synthetic Data Generation and Model-Based Compression

The GenTT framework enables the on-demand generation of long synthetic time series. For instance, in the instant payment scenario, site reliability engineers can generate specific temporal patterns, such as expected transaction volumes for particular days, to validate system capacity and load balancing strategies.

Additionally, the model could serve as a compact representation of the data for model-based compression. Rather than storing large historical datasets, the trained generator could be deployed to generate realistic observations on demand. This is particularly valuable for distributed load testing: lightweight models produce traffic patterns locally, reducing storage needs and network bandwidth consumption while maintaining original statistical properties.

Music Generation and Loopable Audio Synthesis

The capability of GenTT to generate periodic patterns could be extended to music generation. The musical instrument digital interface (MIDI) format could be treated as a multivariate time series where each channel represents a different instrument. In this context, representations such as HTE would capture the periodicity inherent in musical loops. This approach would enable two distinct applications: generating loopable music where patterns repeat with subtle variations, and extending musical sequences while maintaining harmonic consistency.

Part VI

APPENDIX

Your turn to generate.

a

SOFTWARE IMPLEMENTATION

This appendix details the implementation of the GenTT framework for seasonal time series forecasting and anomaly detection.

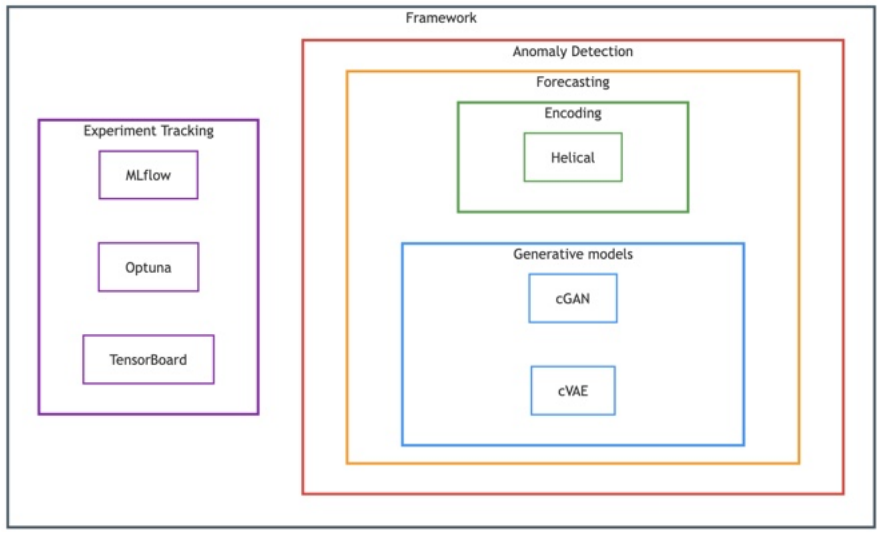


Figure a.1: Architecture of the GenTT framework. Core generative models and time encodings support time series synthesis for forecasting and anomaly detection tasks. Experiment tracking tools provide infrastructure for model development and evaluation.

As illustrated in Figure a.1, the codebase is organized into hierarchical components based on functionality. At the core is the time encoding layer implementing HTE, which serves as input to the cGAN and cVAE. These models support both forecasting and anomaly detection tasks; the latter extends the forecasting architecture. The implementation relies on TensorFlow [400], leveraging computational graphs and GPU acceleration for training.

The experiment tracking system integrates MLflow [275] for logging model parameters, metrics, and artifacts; Optuna for hyper-

parameter optimization; and TensorBoard [399] for computational graph inspection and training visualization. This integration enables reproducible experimentation and systematic comparison across configurations and datasets.

Future extensions include additional generative architectures such as diffusion-based models, alternative temporal encoding schemes (e.g., toroidal and polar representations), and support for tasks beyond forecasting and anomaly detection.

b

INTEGRATION WITH MONITORING INFRASTRUCTURE

This appendix details the integration of the GenTT framework into the Elastic stack [17] for real-time anomaly detection in the TIPS infrastructure. The proposed architecture combines an ETL pipeline for telemetry ingestion, a GPU-enabled environment for model training, a centralized tracking server for model management, and Kibana [197] for visualization.

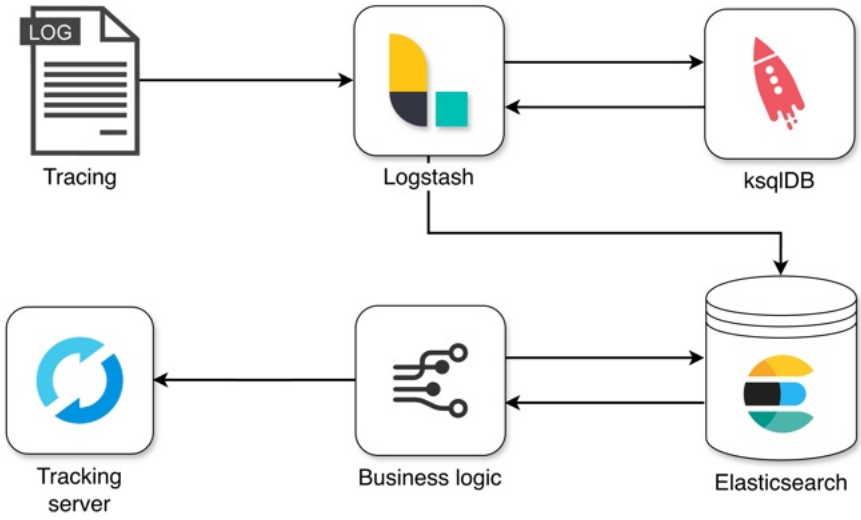


Figure b.1: Integration architecture of the GenTT framework within a monitoring infrastructure. Data flows from tracing logs through the ETL pipeline (Logstash, ksqlDB) to Elasticsearch. The business logic component handles model training and inference, interacts with the tracking server for model lifecycle management, and pushes predictions back to Elasticsearch.

Data flow begins with the streaming extraction of ISO 20022 tracing data from distributed application nodes, as illustrated in Figure b.1.

Source-level filtering and Logstash [272] preprocess the data, while ksqlDB [491] handles complex message correlation to reconstruct complete transactions (Chapter 9). Reconstructed transactions are enriched with derived features such as processing times ($\delta_1, \delta_2, \delta_3$) and indexed in Elasticsearch, enabling both analysis and model training.

Model training occurs within a dedicated GPU environment interfacing with a central tracking server (MLflow [275]). This server logs experiment metadata including parameters, metrics, and model lineage; stores trained model binaries in an artifact repository; and provides an interface for experiment comparison and model registry management. The training module periodically queries Elasticsearch for recent historical data to retrain or fine-tune the GenTT model, adapting to evolving seasonal patterns. Training metadata is logged to the tracking server, and validated models are tagged and deployed to the registry for production inference.

Predictions are written to local files, read by Filebeat, and forwarded to an Elasticsearch index. Anomaly scores are computed in real time by comparing predictions against actual observations. This integration leverages native Elasticsearch anomaly detection capabilities by applying dynamic thresholding directly to calculated anomaly scores and utilizing built-in alerting functionality. Kibana dashboards visualize results in real time, allowing operators to monitor system behavior and assess detected anomalies.

PUBLICATIONS

The complete list of publications appears below. An asterisk (∗) denotes publications discussed in this dissertation.

INTERNATIONAL JOURNAL PAPERS

J1. **L. Porcelli**, M. Ficco, G. D'Angelo, and F. Palmieri. *Context-Aware Coverage Path Planning for a Swarm of UAVs Using Mobile Ground Stations for Battery-Swapping.* Soft Computing.

J2. **L. Porcelli**, U. Fiore, and F. Palmieri. *Generative Models with Helical Time Encoding for Seasonal Time Series Forecasting.* Engineering Applications of Artificial Intelligence. ∗ [324]

J3. **L. Porcelli**, M. Mastroianni, M. Ficco, and F. Palmieri. *A User-Centered Privacy Policy Management System for Automatic Consent on Cookie Banners.* Computers.

J4. **L. Porcelli**, M. Trovati, and Francesco Palmieri. *Real-Time Anomaly Detection in Seasonal Time Series with Conditional Variational Autoencoder.* Applied Soft Computing. ∗ [325]

INTERNATIONAL CONFERENCE PAPERS

C1. P. Fusco, **L. Porcelli**, F. Palmieri, and M. Ficco. *Supporting UAVs Swarm Missions by Multi-Agent Reinforcement Learning.* International Conference on Computing, Networking and Communications (ICNC). 2025.

C2. **L. Porcelli**, M. Ficco and F. Palmieri. *Mitigating User Exposure to Dark Patterns in Cookie Banners Through Automated Consent.* International Conference on Computational Science and Its Applications (ICCSA). 2023.

C3. **L. Porcelli** and F. Palmieri. *Raise Awareness of the Environmental Impacts of Retail Food Products: A User-Centered Scenario-Based*

Approach. International Conference on Ubiquitous Computing & Ambient Intelligence (UCAmI). 2023

PREPRINTS

P1. **L. Porcelli**. *Business-Aware Failure Detection in Instant Payment Infrastructures.* arXiv preprint. ∗ [322]

OTHER PUBLICATIONS

D1. **L. Porcelli**. *The Evolution of IT Operations in TIPS: From Monitoring to AI-Driven Anomaly Detection.* Markets, Infrastructures, Payment Systems. Banca d'Italia. ∗ [323]

BIBLIOGRAPHY

[1] ACI Worldwide - GlobalData. *It's prime time for real-time*. Tech. rep. ACI Worldwide, 2023.

[2] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." In: *arXiv preprint arXiv:1603.04467* (2016).

[3] Charu C. Aggarwal. *Outlier Analysis*. Springer International Publishing, 2017. ISBN: 9783319475783. DOI: 10.1007/978-3-319-47578-3. URL: http://dx.doi.org/10.1007/978-3-319-47578-3.

[4] Subutai Ahmad, Alexander Lavin, Scott Purdy, and Zuha Agha. "Unsupervised real-time anomaly detection for streaming data." In: *Neurocomputing* 262 (2017), pp. 134–147.

[5] Nesreen K Ahmed, Amir F Atiya, Neamat El Gayar, and Hisham El-Shishiny. "An empirical comparison of machine learning models for time series forecasting." In: *Econometric reviews* 29.5-6 (2010), pp. 594–621.

[6] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. "Internet of things: A survey on enabling technologies, protocols, and applications." In: *IEEE communications surveys & tutorials* 17.4 (2015), pp. 2347–2376.

[7] Talayeh Aledavood, Eduardo López, Sam GB Roberts, Felix Reed-Tsochas, Esteban Moro, Robin IM Dunbar, and Jari Saramäki. "Daily rhythms in mobile telephone communication." In: *PloS one* 10.9 (2015), e0138098. URL: https://doi.org/10.1371/journal.pone.0138098.

[8] Morteza Alizadeh, Michael Hamilton, Parker Jones, Junfeng Ma, and Raed Jaradat. "Vehicle operating state anomaly detection and results virtual reality interpretation." In: *Expert Systems with Applications* 177 (2021), p. 114928.

[9] Torben G Andersen, Tim Bollerslev, Peter Christoffersen, and Francis X Diebold. *Volatility forecasting*. 2005.

[10] Sardar Ansari, Negar Farzaneh, Marlena Duda, Kelsey Horan, Hedvig B Andersson, Zachary D Goldberger, Brahmajee K Nallamothu, and Kayvan Najarian. "A review of automated methods for detection of myocardial ischemia and infarction using electrocardiogram and electronic health records." In: *IEEE reviews in biomedical engineering* 10 (2017), pp. 264–298.

[11] Mauro Arcese, Domenico Di Giulio, and Vitangelo Lasorella. "Real-Time Gross Settlement systems: breaking the wall of scalability and high availability." In: *Markets, Infrastructures, Payment Systems* 2 (2021). URL: https://www.bancaditalia.it/pubblicazioni/mercati-infrastrutture-e-sistemi-di-pagamento/approfondimenti/2021-002/index.html.

[12] Ron Artstein and Massimo Poesio. "Inter-Coder Agreement for Computational Linguistics." In: *Computational Linguistics* 34.4 (Dec. 2008), pp. 555–596. ISSN: 1530-9312. DOI: 10.1162/coli.07-034-r2. URL: http://dx.doi.org/10.1162/coli.07-034-r2.

[13] Julien Audibert. "Unsupervised anomaly detection in time-series." PhD thesis. Sorbonne Université, 2021.

[14] 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 international conference on knowledge discovery & data mining*. 2020, pp. 3395–3404.

[15] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, and Maria A Zuluaga. "Do deep neural networks contribute to multivariate time series anomaly detection?" In: *Pattern Recognition* 132 (2022), p. 108945.

[16] Joseph Azar, Abdallah Makhoul, Raphaël Couturier, and Jacques Demerjian. "Robust IoT time series classification with data compression and deep learning." In: *Neurocomputing* 398 (2020), pp. 222–234.

[17] Elasticsearch B.V. *The Elastic Stack*. https://www.elastic.co/docs/get-started/the-stack.

[18] Anton Babenko, Leonardo Mariani, and Fabrizio Pastore. "Ava: automated interpretation of dynamically detected anomalies." In: *Proceedings of the eighteenth international symposium on Software testing and analysis*. 2009, pp. 237–248.

[19] Vytautas Bagdonavičius and Linas Petkevičius. "Multiple outlier detection tests for parametric models." In: *Mathematics* 8.12 (2020), p. 2156.

[20] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." In: *arXiv preprint arXiv:1409.0473* (2014).

[21] Ningning Bai, Xiaofeng Wang, Ruidong Han, Qin Wang, and Zinian Liu. "PAFormer: Anomaly Detection of Time Series With Parallel-Attention Transformer." In: *IEEE Transactions on Neural Networks and Learning Systems* 36.2 (Feb. 2025), 3315–3328. ISSN: 2162-2388. DOI: 10.1109/tnnls.2023.3337876. URL: http://dx.doi.org/10.1109/tnnls.2023.3337876.

[22] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling." In: *arXiv preprint arXiv:1803.01271* (2018). URL: https://doi.org/10.48550/arXiv.1803.01271.

[23] Banca d'Italia. *Access of Western Balkan central banks to TARGET Instant Payment Settlement (TIPS)*. https://www.bancaditalia.it/media/notizia/access-of-western-balkan-central-banks-to-target-instant-payment-settlement-tips/?com.dotmarketing.htmlpage.language=1&dotcache=refresh. 2024.

[24] Kasun Bandara, Hansika Hewamalage, Yuan-Hao Liu, Yanfei Kang, and Christoph Bergmeir. "Improving the accuracy of global forecasting models using time series data augmentation." In: *Pattern Recognition* 120 (2021), p. 108148.

[25] Dor Bank, Noam Koenigstein, and Raja Giryes. "Autoencoders." In: *Machine learning for data science handbook: data mining and knowledge discovery handbook* (2023), pp. 353–374.

[26] Oresti Banos, Juan-Manuel Galvez, Miguel Damas, Hector Pomares, and Ignacio Rojas. "Window size impact in human activity recognition." In: *Sensors* 14.4 (2014), pp. 6474–6499.

[27] Alberto Barbado. "Anomaly detection in average fuel consumption with XAI techniques for dynamic generation of explanations." In: *ArXiv abs/2010.16051* (2020).

[28] Alberto Barbado, Óscar Corcho, and Richard Benjamins. "Rule extraction in unsupervised anomaly detection for model explainability: Application to OneClass SVM." In: *Expert Systems with Applications* 189 (2022), p. 116100.

[29] Md Abul Bashar and Richi Nayak. "TAnoGAN: Time series anomaly detection with generative adversarial networks." In: *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE. 2020, pp. 1778–1785.

[30] Jonathan Baxter. "A model of inductive bias learning." In: *Journal of artificial intelligence research* 12 (2000), pp. 149–198.

[31] Inci M. Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K. Jain, and Jiayu Zhou. "Patient Subtyping via Time-Aware LSTM Networks." In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '17. ACM, Aug. 2017, pp. 65–74. DOI: 10.1145/3097983.3097997. URL: http://dx.doi.org/10.1145/3097983.3097997.

[32] Katharina Beckh, Sebastian Müller, Matthias Jakobs, Vanessa Toborek, Hanxiao Tan, Raphael Fischer, Pascal Welke, Sebastian Houben, and Laura von Rueden. "Explainable machine learning with prior knowledge: an overview." In: *arXiv preprint arXiv:2105.10172* (2021).

[33] Souhaib Ben Taieb, Gianluca Bontempi, Amir F. Atiya, and Antti Sorjamaa. "A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition." en. In: *Expert Systems with Applications* 39.8 (June 2012), pp. 7067–7083. ISSN: 09574174. DOI: 10.1016/j.eswa.2012.01.039. URL: https://linkinghub.elsevier.com/retrieve/pii/S0957417412000528 (visited on 10/02/2025).

[34] Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation learning: A review and new perspectives." In: *IEEE transactions on pattern analysis and machine intelligence* 35.8 (2013), pp. 1798–1828.

[35] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult." In: *IEEE transactions on neural networks* 5.2 (1994), pp. 157–166.

[36] Betsy Beyer, Chris Jones, Jennifer Petoff, and Niall Richard Murphy. *Site reliability engineering: How Google runs production systems*. " O'Reilly Media, Inc.", 2016.

[37] A. M. Bianco, M. García Ben, E. J. Martínez, and V. J. Yohai. "Outlier Detection in Regression Models with ARIMA Errors using Robust Estimates." In: *Journal of Forecasting* 20.8 (Dec. 2001), 565–579. ISSN: 1099-131X. DOI: 10.1002/for.768. URL: http://dx.doi.org/10.1002/for.768.

[38] Christopher M Bishop. "Novelty detection and neural network validation." In: *IEE Proceedings-Vision, Image and Signal processing* 141.4 (1994), pp. 217–222.

[39] Ane Blázquez-García, Angel Conde, Usue Mori, and Jose A Lozano. "A review on outlier/anomaly detection in time series data." In: *ACM computing surveys (CSUR)* 54.3 (2021), pp. 1–33.

[40] Avrim Blum. "Empirical Support for Winnow and Weighted-Majority Algorithms: Results on a Calendar Scheduling Domain." In: *Machine Learning* 26.1 (Jan. 1997), pp. 5–23. ISSN: 1573-0565. DOI: 10.1023/a:1007335615132. URL: http://dx.doi.org/10.1023/a:1007335615132.

[41] Paul Boniol, Ashwin K. Krishna, Marine Bruel, Qinghua Liu, Mingyi Huang, Themis Palpanas, Ruey S. Tsay, Aaron Elmore, Michael J. Franklin, and John Paparrizos. "VUS: effective and efficient accuracy measures for time-series anomaly detection." In: *The VLDB Journal* 34.3 (Mar. 2025). ISSN: 0949-877X. DOI: 10.1007/s00778-025-00907-x. URL: http://dx.doi.org/10.1007/s00778-025-00907-x.

[42] Ali Borji. "Pros and cons of GAN evaluation measures." In: *Computer vision and image understanding* 179 (2019), pp. 41–65.

[43] Anastasia Borovykh, Sander Bohte, and Cornelis W Oosterlee. "Conditional time series forecasting with convolutional neural networks." In: *arXiv preprint arXiv:1703.04691* (2017).

[44] Joos-Hendrik Böse, Valentin Flunkert, Jan Gasthaus, Tim Januschowski, Dustin Lange, David Salinas, Sebastian Schelter, Matthias Seeger, and Yuyang Wang. "Probabilistic demand forecasting at scale." In: *Proceedings of the VLDB Endowment* 10.12 (2017), pp. 1694–1705.

[45] G. E. P. Box and G. M. Jenkins. "Some Recent Advances in Forecasting and Control." In: *Applied Statistics* 17.2 (1968), p. 91. ISSN: 0035-9254. DOI: 10.2307/2985674. URL: http://dx.doi.org/10.2307/2985674.

[46] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[47] Gustav Bredell, Kyriakos Flouris, Krishna Chaitanya, Ertunc Erdil, and Ender Konukoglu. "Explicitly minimizing the blur error of variational autoencoders." In: *arXiv preprint arXiv:2304.05939* (2023).

[48] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. "LOF: identifying density-based local outliers." In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. 2000, pp. 93–104.

[49] Andrew Brock. "Large Scale GAN Training for High Fidelity Natural Image Synthesis." In: *arXiv preprint arXiv:1809.11096* (2018).

[50] Eoin Brophy, Zhengwei Wang, Qi She, and Tomás Ward. "Generative Adversarial Networks in Time Series: A Systematic Literature Review." In: *ACM Computing Surveys* 55.10 (Feb. 2023), pp. 1–31. ISSN: 1557-7341. DOI: 10.1145/3559540. URL: http://dx.doi.org/10.1145/3559540.

[51] Andy Brown, Aaron Tuor, Brian Hutchinson, and Nicole Nichols. "Recurrent neural network attention mechanisms for interpretable system log anomaly detection." In: *Proceedings of the First Workshop on Machine Learning for Computing Systems*. 2018, pp. 1–8.

[52] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners." In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901. URL: https://doi.org/10.48550/arXiv.2005.14165.

[53] Andriy Burkov. *Machine learning engineering*. Vol. 1. True Positive Incorporated Montreal, QC, Canada, 2020.

[54] Giorgio Buttazzo. *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*. Springer Nature Switzerland, 2024. ISBN: 9783031454103. DOI: 10.1007/978-3-031-45410-3. URL: http://dx.doi.org/10.1007/978-3-031-45410-3.

[55] David Campos, Tung Kieu, Chenjuan Guo, Feiteng Huang, Kai Zheng, Bin Yang, and Christian S. Jensen. "Unsupervised time series outlier detection with diversity-driven convolutional ensembles." In: *Proceedings of the VLDB Endowment* 15.3 (Nov. 2021), pp. 611–623. ISSN: 2150-8097. DOI: 10.14778/3494124.3494142. URL: http://dx.doi.org/10.14778/3494124.3494142.

[56] Gianluca Caricato, Marco Capotosto, Silvio Orsini, and Pietro Tiberi. "TIPS: A Zero-Downtime Platform Powered by Automation." In: *Markets, Infrastructures, Payment Systems* 28 (2022). URL: https://www.bancaditalia.it/pubblicazioni/

mercati - infrastrutture - e - sistemi - di - pagamento / approfondimenti/2022-028/index.html.

[57] Ander Carreño, Iñaki Inza, and Jose A. Lozano. "Analyzing rare event, anomaly, novelty and outlier detection terms under the supervised classification framework." In: *Artificial Intelligence Review* 53.5 (Oct. 2019), pp. 3575–3594. ISSN: 1573-7462. DOI: 10.1007/s10462-019-09771-y. URL: http://dx.doi.org/10.1007/s10462-019-09771-y.

[58] Debaditya Chakraborty and Hazem Elzarka. "Advanced machine learning techniques for building performance simulation: a comparative analysis." In: *Journal of Building Performance Simulation* 12.2 (2019), pp. 193–207. URL: https://doi.org/10.1080/19401493.2018.1498538.

[59] Thanyalak Chalermarrewong, Tiranee Achalakul, and Simon Chong Wee See. "Failure Prediction of Data Centers Using Time Series and Fault Tree Analysis." In: *2012 IEEE 18th International Conference on Parallel and Distributed Systems*. IEEE, Dec. 2012, 794–799. DOI: 10.1109/icpads.2012.129. URL: http://dx.doi.org/10.1109/icpads.2012.129.

[60] Varun Chandola, Arindam Banerjee, and Vipin Kumar. "Anomaly detection: A survey." In: *ACM computing surveys (CSUR)* 41.3 (2009), pp. 1–58.

[61] Chengcheng Chen, Qian Zhang, M. Kashani, C. Jun, S. M. Bateni, Shahab S. Band, S. S. Dash, and K. Chau. "Forecast of rainfall distribution based on fixed sliding window long short-term memory." In: *Engineering Applications of Computational Fluid Mechanics* 16 (2022), pp. 248–261. DOI: 10.1080/19942060.2021.2009374.

[62] Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O Arik, and Tomas Pfister. "Tsmixer: An all-mlp architecture for time series forecasting." In: *arXiv preprint arXiv:2303.06053* (2023).

[63] Tianqi Chen and Carlos Guestrin. "XGBoost: A Scalable Tree Boosting System." In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. ACM, Aug. 2016, pp. 785–794. DOI: 10.1145/293

9672.2939785. URL: http://dx.doi.org/10.1145/2939672.2939785.

[64] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. "A simple framework for contrastive learning of visual representations." In: *International conference on machine learning*. PmLR. 2020, pp. 1597–1607.

[65] Xi Chen, Yateng Tang, Jiarong Xu, Jiawei Zhang, Siwei Zhang, Sijia Peng, Xuehao Zheng, and Yun Xiong. *Rethinking Time Encoding via Learnable Transformation Functions*. arXiv:2505.00887 [cs]. May 2025. DOI: 10.48550/arXiv.2505.00887. URL: http://arxiv.org/abs/2505.00887.

[66] Xuanhao Chen, Liwei Deng, Feiteng Huang, Chengwei Zhang, Zongquan Zhang, Yan Zhao, and Kai Zheng. "Daemon: Unsupervised anomaly detection and interpretation for multivariate time series." In: *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE. 2021, pp. 2225–2230.

[67] Yinfang Chen, Manish Shetty, Gagan Somashekar, Minghua Ma, Yogesh Simmhan, Jonathan Mace, Chetan Bansal, Rujia Wang, and Saravan Rajmohan. "AIOpsLab: A Holistic Framework to Evaluate AI Agents for Enabling Autonomous Clouds." In: *arXiv preprint arXiv:2501.06706* (2025).

[68] Yinfang Chen et al. "Automatic Root Cause Analysis via Large Language Models for Cloud Incidents." In: *Proceedings of the Nineteenth European Conference on Computer Systems*. EuroSys '24. ACM, Apr. 2024, pp. 674–688. DOI: 10.1145/3627703.3629553. URL: http://dx.doi.org/10.1145/3627703.3629553.

[69] Yushu Chen, Shengzhuo Liu, Jinzhe Yang, Hao Jing, Wenlai Zhao, and Guang-Wu Yang. "A Joint Time-frequency Domain Transformer for Multivariate Time Series Forecasting." In: *Neural networks : the official journal of the International Neural Network Society* 176 (2023), p. 106334. DOI: 10.48550/arXiv.2305.14649.

[70] Dawei Cheng, Fangzhou Yang, Sheng Xiang, and Jin Liu. "Financial time series forecasting with multi-modality graph

neural network." In: *Pattern Recognition* 121 (2022), p. 108218. URL: https://doi.org/10.1016/j.patcog.2021.108218.

[71] Chiara Scotti. *From Magma to Masterpiece: Forging the Future of Cross-border Payments*. https://www.bancaditalia.it/pubblicazioni/interventi-direttorio/int-dir-2025/Scotti-08.05.2025.pdf. Keynote speech, Reykjavík Economic Conference. 2025.

[72] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." In: *arXiv preprint arXiv:1406.1078* (2014).

[73] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F Stewart, and Jimeng Sun. "Doctor ai: Predicting clinical events via recurrent neural networks." In: *Machine learning for healthcare conference*. PMLR. 2016, pp. 301–318. URL: https://doi.org/10.48550/arXiv.1511.05942.

[74] Jeongwhan Choi, Hwangyong Choi, Jeehyun Hwang, and Noseong Park. "Graph neural controlled differential equations for traffic forecasting." In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 36. 6. 2022, pp. 6367–6374.

[75] Kukjin Choi, Jihun Yi, Changhwa Park, and Sungroh Yoon. "Deep learning for anomaly detection in time-series data: Review, analysis, and guidelines." In: *IEEE access* 9 (2021), pp. 120043–120065.

[76] Seung-Hwan Choi, Dawn An, Inho Lee, and Suwoong Lee. "Anomaly Detection Based on Graph Convolutional Network–Variational Autoencoder Model Using Time-Series Vibration and Current Data." In: *Mathematics* 12.23 (Nov. 2024), p. 3750. ISSN: 2227-7390. DOI: 10.3390/math12233750. URL: http://dx.doi.org/10.3390/math12233750.

[77] Yeji Choi, Hyunki Lim, Heeseung Choi, and Ig-Jae Kim. "Gan-based anomaly detection and localization of multivariate time series data for power plant." In: *2020 IEEE international con-*

ference on big data and smart computing (BigComp). IEEE. 2020, pp. 71–74.

[78] Yinghao Chu, Hugo T.C. Pedro, and Carlos F.M. Coimbra. "Hybrid intra-hour DNI forecasts with sky image processing enhanced by stochastic learning." In: *Solar Energy* 98 (Dec. 2013), pp. 592–603. ISSN: 0038-092X. DOI: 10.1016/j.solener.2013.10.020. URL: http://dx.doi.org/10.1016/j.solener.2013.10.020.

[79] City of New York. *Taxi and Limousine Commission Trip Record Data*. https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page. Accessed: 2025-09-01.

[80] EBA Clearing. *RT1*. URL: https://www.ebaclearing.eu/services/rt1/overview/ (visited on 10/29/2024).

[81] Robert B Cleveland, William S Cleveland, Jean E McRae, Irma Terpenning, et al. "STL: A seasonal-trend decomposition." In: *J. off. Stat* 6.1 (1990), pp. 3–73.

[82] Jacob Cohen. "A Coefficient of Agreement for Nominal Scales." In: *Educational and Psychological Measurement* 20.1 (Apr. 1960), pp. 37–46. ISSN: 1552-3888. DOI: 10.1177/001316446002000104. URL: http://dx.doi.org/10.1177/001316446002000104.

[83] Committee on Payment and Settlement Systems - Technical Committee of the International Organization of Securities Commissions. *Principles for financial market infrastructures*. Tech. rep. Bank for International Settlements and International Organization of Securities Commissions, 2012.

[84] Committee on Payment and Settlement Systems of the central banks of the Group of Ten countries. *Real-Time Gross Settlement Systems*. Tech. rep. Bank for International Settlements, 1997.

[85] Andrew A. Cook, Goksel Misirli, and Zhong Fan. "Anomaly Detection for IoT Time-Series Data: A Survey." en. In: *IEEE Internet of Things Journal* 7.7 (July 2020), pp. 6481–6494. ISSN: 2327-4662, 2372-2541. DOI: 10.1109/JIOT.2019.2958185. URL: https://ieeexplore.ieee.org/document/8926446/ (visited on 09/22/2025).

[86] European Payments Council. *Clearing and Settlement Mechanisms*. URL: https://www.europeanpaymentscouncil.eu/what-we-do/epc-payment-scheme-management/clearing-and-settlement-mechanisms (visited on 10/29/2024).

[87] European Payments Council. *SEPA Instant Credit Transfer*. URL: https://www.europeanpaymentscouncil.eu/what-we-do/sepa-instant-credit-transfer (visited on 10/29/2024).

[88] Enyan Dai and Jie Chen. "Graph-augmented normalizing flows for anomaly detection of multiple time series." In: *arXiv preprint arXiv:2202.07857* (2022).

[89] Yingnong Dang, Qingwei Lin, and Peng Huang. "Aiops: real-world challenges and research innovations." In: *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*. IEEE. 2019, pp. 4–5.

[90] Zahra Zamanzadeh Darban, Yiyuan Yang, Geoffrey I Webb, Charu C Aggarwal, Qingsong Wen, Shirui Pan, and Mahsa Salehi. "DACAD: Domain adaptation contrastive learning for anomaly detection in multivariate time series." In: *IEEE Transactions on Knowledge and Data Engineering* (2025).

[91] Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. "Long-term forecasting with tide: Time-series dense encoder." In: *arXiv preprint arXiv:2304.08424* (2023).

[92] Arun Das and Paul Rad. "Opportunities and challenges in explainable artificial intelligence (xai): A survey." In: *arXiv preprint arXiv:2006.11371* (2020).

[93] Alysha M. De Livera, Rob J. Hyndman, and Ralph D. Snyder. "Forecasting Time Series With Complex Seasonal Patterns Using Exponential Smoothing." In: *Journal of the American Statistical Association* 106.496 (Dec. 2011), pp. 1513–1527. ISSN: 1537-274X. DOI: 10.1198/jasa.2011.tm09771. URL: http://dx.doi.org/10.1198/jasa.2011.tm09771.

[94] Chirag Deb, Fan Zhang, Junjing Yang, Siew Eang Lee, and Kwok Wei Shah. "A review on time series forecasting techniques for building energy consumption." In: *Renewable and Sustainable Energy Reviews* 74 (2017), pp. 902–924.

[95] Ailin Deng and Bryan Hooi. "Graph neural network-based anomaly detection in multivariate time series." In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 5. 2021, pp. 4027–4035.

[96] Marie Devaine, Pierre Gaillard, Yannig Goude, and Gilles Stoltz. "Forecasting electricity consumption by aggregating specialized experts: A review of the sequential aggregation of specialized experts, with an application to Slovakian and French country-wide one-day-ahead (half-)hourly predictions." In: *Machine Learning* 90.2 (Aug. 2012), pp. 231–260. ISSN: 1573-0565. DOI: 10.1007/s10994-012-5314-7. URL: http://dx.doi.org/10.1007/s10994-012-5314-7.

[97] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "BERT: Pre-training of deep bidirectional transformers for language understanding." In: *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 2019, pp. 4171–4186. URL: https://doi.org/10.48550/arXiv.1810.04805.

[98] Carl Doersch. "Tutorial on variational autoencoders." In: *arXiv preprint arXiv:1606.05908* (2016).

[99] Finale Doshi-Velez and Been Kim. "Towards a rigorous science of interpretable machine learning." In: *arXiv preprint arXiv:1702.08608* (2017).

[100] Kenji Doya and Shuji Yoshizawa. "Adaptive neural oscillator using continuous-time back-propagation learning." In: *Neural Networks* 2.5 (1989), pp. 375–385.

[101] Bowen Du, Hao Peng, Senzhang Wang, Md Zakirul Alam Bhuiyan, Lihong Wang, Qiran Gong, Lin Liu, and Jing Li. "Deep irregular convolutional residual LSTM for urban traffic passenger flows prediction." In: *IEEE Transactions on Intelligent*

Transportation Systems 21.3 (2019), pp. 972–985. URL: https://doi.org/10.1109/TITS.2019.2900481.

[102] EBA Clearing. *RT1*. https://www.ebaclearing.eu/services/rt1. Accessed: 2024-07-06. 2024.

[103] EPC Ad-hoc Task Force on Instant Payments. *EPC Report on Instant Payments*. Tech. rep. European Payments Council, 2015.

[104] Elasticsearch B.V. *Anomaly Detection - Elastic Documentation*. Accessed: 2024-06-17. 2024. URL: https://www.elastic.co/docs/explore-analyze/machine-learning/anomaly-detection.

[105] Ayman Elhalwagy and Tatiana Kalganova. "Hybridization of Capsule and LSTM Networks for unsupervised anomaly detection on multivariate data." In: *arXiv preprint arXiv:2202.05538* (2022).

[106] Jeffrey L Elman. "Finding structure in time." In: *Cognitive science* 14.2 (1990), pp. 179–211.

[107] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. "Neural architecture search: A survey." In: *Journal of Machine Learning Research* 20.55 (2019), pp. 1–21.

[108] Tolga Ergen and Suleyman Serdar Kozat. "Unsupervised Anomaly Detection With LSTM Neural Networks." In: *IEEE Transactions on Neural Networks and Learning Systems* 31.8 (Aug. 2020), pp. 3127–3141. ISSN: 2162-2388. DOI: 10.1109/tnnls.2019.2935975. URL: http://dx.doi.org/10.1109/tnnls.2019.2935975.

[109] Philippe Esling and Carlos Agon. "Time-series data mining." In: *ACM Computing Surveys (CSUR)* 45.1 (2012), pp. 1–34.

[110] Cristóbal Esteban, Stephanie L Hyland, and Gunnar Rätsch. "Real-valued (medical) time series generation with recurrent conditional gans." In: *arXiv preprint arXiv:1706.02633* (2017).

[111] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *kdd*. Vol. 96. 34. 1996, pp. 226–231.

[112] European Central Bank. *Sweden joins TIPS – Eurosystem instant payments platform also settles in kronor*. URL: https://www.ecb.europa.eu/press/intro/news/html/ecb.mipnews240227.en.html (visited on 02/27/2024).

[113] European Central Bank. *What is TIPS?* URL: https://www.ecb.europa.eu/paym/target/tips/html/index.en.html (visited on 10/29/2024).

[114] European Central Bank. *What is the Single Instructing Party settlement model for instant payments?* URL: https://www.ecb.europa.eu/paym/target/target-professional-use-documents-links/tips/shared/pdf/tips-champions-sip-explanation.pdf (visited on 05/31/2021).

[115] European Central Bank. *TARGET Annual Report 2023*. https://www.ecb.europa.eu/press/targetar/html/ecb.targetar2023.en.html. Accessed: 2024-07-06. 2024.

[116] European Council and Council of the European Union. *Council adopts regulation on instant payments - Consilium*. Accessed: 2024-07-06. 2024. URL: https://www.consilium.europa.eu/en/press/press-releases/2024/02/26/council-adopts-regulation-on-instant-payments/.

[117] European Payments Council. *EPC125-05 SEPA Credit Transfer Scheme*. Rulebook. Version 1.1. European Payments Council, 2025. URL: https://www.europeanpaymentscouncil.eu/document-library/rulebooks/2025-sepa-credit-transfer-rulebook-version-11.

[118] Wei Fan, Kun Yi, Hangting Ye, Zhiyuan Ning, Qi Zhang, and Ning An. "Deep frequency derivative learning for non-stationary time series forecasting." In: *arXiv preprint arXiv:2407.00502* (2024).

[119] Zheng Fang, Qingqing Long, Guojie Song, and Kunqing Xie. "Spatial-temporal graph ode networks for traffic flow forecasting." In: *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 2021, pp. 364–373.

[120] Tom Fawcett. "An introduction to ROC analysis." In: *Pattern Recognition Letters* 27.8 (June 2006), pp. 861–874. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2005.10.010. URL: http://dx.doi.org/10.1016/j.patrec.2005.10.010.

[121] Falih Gozi Febrinanto, Kristen Moore, Chandra Thapa, Mujie Liu, Vidya Saikrishna, Jiangang Ma, and Feng Xia. "Entropy causal graphs for multivariate time series anomaly detection." In: *ACM Transactions on Intelligent Systems and Technology* 16.6 (2025), pp. 1–25.

[122] Binbin Feng, Zhijun Ding, and Changjun Jiang. "FAST: A forecasting model with adaptive sliding window and time locality integration for dynamic cloud workloads." In: *IEEE Transactions on Services Computing* 16.2 (2022), pp. 1184–1197.

[123] Cheng Feng and Pengwei Tian. "Time series anomaly detection for cyber-physical systems via neural system identification and bayesian filtering." In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2021, pp. 2858–2867.

[124] Pavel Filonov, Andrey Lavrentyev, and Artem Vorontsov. "Multivariate industrial time series with cyber-attack simulation: Fault detection using an lstm-based predictive data model." In: *arXiv preprint arXiv:1612.06676* (2016).

[125] Ugo Fiore, Alfredo De Santis, Francesca Perla, Paolo Zanetti, and Francesco Palmieri. "Using generative adversarial networks for improving classification effectiveness in credit card fraud detection." In: *Information Sciences* 479 (2019), pp. 448–455.

[126] Evelyn Fix and J. L. Hodges. *Discriminatory analysis: Nonparametric discrimination: Consistency properties: (471672008-001)*. 1951. DOI: 10.1037/e471672008-001. URL: http://dx.doi.org/10.1037/e471672008-001.

[127] Yoav Freund, Robert E. Schapire, Yoram Singer, and Manfred K. Warmuth. "Using and combining predictors that specialize." In: *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing - STOC '97*. STOC '97. ACM Press,

1997, pp. 334–343. DOI: 10.1145/258533.258616. URL: http://dx.doi.org/10.1145/258533.258616.

[128] Tak-chung Fu. "A review on time series data mining." In: *Engineering Applications of Artificial Intelligence* 24.1 (2011), pp. 164–181.

[129] Kelum Gajamannage, Yonggi Park, and D. Jayathilake. "Real-time forecasting of time series in financial markets using sequentially trained dual-LSTMs." In: *Expert Syst. Appl.* 223 (2023), p. 119879. DOI: 10.1016/j.eswa.2023.119879.

[130] Rong Gao, Wei He, Lingyu Yan, Donghua Liu, Yonghong Yu, and Zhiwei Ye. "Hybrid graph transformer networks for multivariate time series anomaly detection." In: *The Journal of Supercomputing* 80.1 (July 2023), 642–669. ISSN: 1573-0484. DOI: 10.1007/s11227-023-05503-w. URL: http://dx.doi.org/10.1007/s11227-023-05503-w.

[131] Shuai Gao, Yuefei Huang, Jingcheng Han, Guangqian Wang, Meixin Zhang, and Qingsheng Lin. "Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample generation." In: *Journal of Hydrology* 589 (2020), p. 125188. DOI: 10.1016/j.jhydrol.2020.125188.

[132] Everette S Gardner Jr. "Exponential smoothing: The state of the art." In: *Journal of forecasting* 4.1 (1985), pp. 1–28.

[133] Michael S Gashler and Stephen C Ashmore. "Modeling time series data with deep Fourier neural networks." In: *Neurocomputing* 188 (2016), pp. 3–11. URL: https://doi.org/10.1016/j.neucom.2015.01.108.

[134] Alexander Geiger, Dongyu Liu, Sarah Alnegheimish, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. "Tadgan: Time series anomaly detection using generative adversarial networks." In: *2020 ieee international conference on big data (big data)*. IEEE. 2020, pp. 33–43.

[135] Getswish AB. *Swish Incident History*. https://status.swish.nu/history. Accessed: 2025-07-06.

[136] Ramin Ghorbani, Marcel J.T. Reinders, and David M.J. Tax. "PATE: Proximity-Aware Time Series Anomaly Evaluation." In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD '24. ACM, Aug. 2024, pp. 872–883. DOI: 10.1145/3637528.3671971. URL: http://dx.doi.org/10.1145/3637528.3671971.

[137] Ioana Giurgiu and Anika Schumann. "Additive explanations for anomalies detected from multivariate temporal data." In: *Proceedings of the 28th acm international conference on information and knowledge management*. 2019, pp. 2245–2248.

[138] Luke B Godfrey and Michael S Gashler. "Neural decomposition of time-series data for effective generalization." In: *IEEE transactions on neural networks and learning systems* 29.7 (2017), pp. 2973–2985. URL: https://doi.org/10.1109/TNNLS.2017.2709324.

[139] Bruno Gonçalves and José J. Ramasco. "Human dynamics revealed through Web analytics." In: *Physical Review E* 78.2 (Aug. 2008). ISSN: 1550-2376. DOI: 10.1103/physreve.78.026123. URL: http://dx.doi.org/10.1103/physreve.78.026123.

[140] Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*. Vol. 1. 2. MIT press Cambridge, 2016.

[141] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets." In: vol. 27. 2014.

[142] Adam Goodge, Bryan Hooi, See Kiong Ng, and Wee Siong Ng. "Robustness of Autoencoders for Anomaly Detection Under Adversarial Impact." In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. IJCAI-PRICAI-2020. International Joint Conferences on Artificial Intelligence Organization, July 2020, pp. 1244–1250. DOI: 10.24963/ijcai.2020/173. URL: http://dx.doi.org/10.24963/ijcai.2020/173.

[143] Alex Graves. "Supervised sequence labelling." In: *Supervised sequence labelling with recurrent neural networks*. Springer, 2012, pp. 5–13.

[144] Siwei Guan, Zhiwei He, Shenhui Ma, and Mingyu Gao. "Conditional normalizing flow for multivariate time series anomaly detection." In: *ISA Transactions* 143 (Dec. 2023), 231–243. ISSN: 0019-0578. DOI: 10.1016/j.isatra.2023.09.002. URL: http://dx.doi.org/10.1016/j.isatra.2023.09.002.

[145] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. "Improved training of wasserstein gans." In: *Advances in neural information processing systems* 30 (2017).

[146] Hongcheng Guo, Jian Yang, Jiaheng Liu, Liqun Yang, Linzheng Chai, Jiaqi Bai, Junran Peng, Xiaorong Hu, Chao Chen, Dongfeng Zhang, et al. "Owl: A large language model for it operations." In: *arXiv preprint arXiv:2309.09298* (2023).

[147] Yifan Guo, Weixian Liao, Qianlong Wang, Lixing Yu, Tianxi Ji, and Pan Li. "Multidimensional time series anomaly detection: A gru-based gaussian mixture variational autoencoder approach." In: *Asian Conference on Machine Learning*. PMLR. 2018, pp. 97–112.

[148] R. Hadsell, S. Chopra, and Y. LeCun. "Dimensionality Reduction by Learning an Invariant Mapping." In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*. Vol. 2. IEEE, 1735–1742. DOI: 10.1109/cvpr.2006.100. URL: http://dx.doi.org/10.1109/cvpr.2006.100.

[149] Pouya Hamadanian, Behnaz Arzani, Sadjad Fouladi, Siva Kesava Reddy Kakarla, Rodrigo Fonseca, Denizcan Billor, Ahmad Cheema, Edet Nkposong, and Ranveer Chandra. "A Holistic View of AI-driven Network Incident Management." In: *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks*. HotNets '23. ACM, Nov. 2023, pp. 180–188. DOI: 10.1145/3626111.3628176. URL: http://dx.doi.org/10.1145/3626111.3628176.

[150] Greg Hamerly, Charles Elkan, et al. "Bayesian approaches to failure prediction for disk drives." In: *ICML*. Vol. 1. 2001. Citeseer. 2001, pp. 202–209.

[151] James D Hamilton. *Time series analysis*. Princeton university press, 2020.

[152] Wolfgang K Härdle, Nikolaus Hautsch, and Andrija Mihoci. "Local adaptive multiplicative error models for high-frequency forecasts." In: *Journal of Applied Econometrics* 30.4 (2015), pp. 529–550.

[153] Andrew C Harvey. "Forecasting, structural time series models and the Kalman filter." In: (1990).

[154] Ali Jameel Hashim, M. A. Balafar, Jafar Tanha, and Aryaz Baradarani. "AEVAE: Adaptive Evolutionary Autoencoder for Anomaly Detection in Time Series." In: *IEEE Transactions on Neural Networks and Learning Systems* 36.1 (Jan. 2025), pp. 1495–1506. ISSN: 2162-2388. DOI: 10.1109/tnnls.2023.3337243. URL: http://dx.doi.org/10.1109/tnnls.2023.3337243.

[155] D. M. Hawkins. *Identification of Outliers*. Springer Netherlands, 1980. ISBN: 9789401539944. DOI: 10.1007/978-94-015-3994-4. URL: http://dx.doi.org/10.1007/978-94-015-3994-4.

[156] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. "Momentum contrast for unsupervised visual representation learning." In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 9729–9738.

[157] Zilong He, Pengfei Chen, and Tao Huang. "Share or Not Share? Towards the Practicability of Deep Models for Unsupervised Anomaly Detection in Modern Online Systems." In: *2022 IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, Oct. 2022, 25–35. DOI: 10.1109/issre55969.2022.00014. URL: http://dx.doi.org/10.1109/issre55969.2022.00014.

[158] Zilong He, Pengfei Chen, Xiaoyun Li, Yongfeng Wang, Guangba Yu, Cailin Chen, Xinrui Li, and Zibin Zheng. "A spatiotemporal deep learning approach for unsupervised

anomaly detection in cloud systems." In: *IEEE Transactions on Neural Networks and Learning Systems* 34.4 (2020), pp. 1705–1719.

[159] Julien Herzen et al. "Darts: User-Friendly Modern Machine Learning for Time Series." In: *Journal of Machine Learning Research* 23.124 (2022), pp. 1–6. URL: http://jmlr.org/papers/v23/21-1177.html.

[160] Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. "Recurrent neural networks for time series forecasting: Current status and future directions." In: *International Journal of Forecasting* 37.1 (2021), pp. 388–427.

[161] Waleed Hilal, S Andrew Gadsden, and John Yawney. "Financial fraud: a review of anomaly detection techniques and recent advances." In: *Expert systems With applications* 193 (2022), p. 116429.

[162] Henrique Steinherz Hippert, Carlos Eduardo Pedreira, and Reinaldo Castro Souza. "Neural networks for short-term load forecasting: A review and evaluation." In: *IEEE Transactions on power systems* 16.1 (2002), pp. 44–55.

[163] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models." In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851. URL: https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

[164] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, Jürgen Schmidhuber, et al. *Gradient flow in recurrent nets: the difficulty of learning long-term dependencies*. 2001.

[165] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory." In: *Neural computation* 9.8 (1997), pp. 1735–1780. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[166] Arthur E Hoerl and Robert W Kennard. "Ridge regression: applications to nonorthogonal problems." In: *Technometrics* (1970), pp. 69–82. URL: https://doi.org/10.2307/1267352.

[167] Hadi Hojjati, Thi Kieu Khanh Ho, and Narges Armanfard. "Self-supervised anomaly detection in computer vision and beyond: A survey and outlook." In: *Neural Networks* 172 (Apr. 2024), p. 106106. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2024.106106. URL: http://dx.doi.org/10.1016/j.neunet.2024.106106.

[168] Huiyue Huang, Lei Yang, Yuanbin Wang, Xun Xu, and Yuqian Lu. "Digital twin-driven online anomaly detection for an automation system based on edge intelligence." In: *Journal of Manufacturing Systems* 59 (2021), pp. 138–150.

[169] Xiaoqiao Huang, Qiong Li, Yonghang Tai, Zaiqing Chen, Jun Liu, Junsheng Shi, and Wuming Liu. "Time series forecasting for hourly photovoltaic power using conditional generative adversarial network and Bi-LSTM." In: *Energy* 246 (2022), p. 123403. URL: https://doi.org/10.1016/j.energy.2022.123403.

[170] Xiaoyu Huang, Weidong Chen, Bo Hu, and Zhendong Mao. "Graph mixture of experts and memory-augmented routers for multivariate time series anomaly detection." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 39. 16. 2025, pp. 17476–17484.

[171] Alexis Huet, Jose Manuel Navarro, and Dario Rossi. "Local Evaluation of Time Series Anomaly Detection Algorithms." In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD '22. ACM, Aug. 2022, pp. 635–645. DOI: 10.1145/3534678.3539339. URL: http://dx.doi.org/10.1145/3534678.3539339.

[172] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding." In: *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 2018, pp. 387–395.

[173] Truong Thu Huong, Ta Phuong Bac, Dao Minh Long, Tran Duc Luong, Nguyen Minh Dan, Bui Doan Thang, Kim Phuc Tran, et al. "Detecting cyberattacks using anomaly detection in industrial control systems: A federated learning approach." In: *Computers in Industry* 132 (2021), p. 103509.

[174] Won-Seok Hwang, Jeong-Han Yun, Jonguk Kim, and Hyoung Chun Kim. "Time-Series Aware Precision and Recall for Anomaly Detection: Considering Variety of Detection Result and Addressing Ambiguous Labeling." In: *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. CIKM '19. ACM, Nov. 2019, pp. 2241–2244. DOI: 10.1145/3357384.3358118. URL: http://dx.doi.org/10.1145/3357384.3358118.

[175] Rob J Hyndman. "A brief history of forecasting competitions." In: *International Journal of Forecasting* 36.1 (2020), pp. 7–14.

[176] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[177] Boris Iglewicz and David C. Hoaglin. *How to Detect and Handle Outliers*. Vol. 16. The ASQC basic references in quality control: statistical techniques. ASQC Quality Press, 1993, pp. 1–87.

[178] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. "Adaptive Mixtures of Local Experts." In: *Neural Computation* 3.1 (Feb. 1991), pp. 79–87. ISSN: 1530-888X. DOI: 10.1162/neco.1991.3.1.79. URL: http://dx.doi.org/10.1162/neco.1991.3.1.79.

[179] Herbert Jaeger. "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note." In: *Bonn, Germany: German national research center for information technology gmd technical report* 148.34 (2001), p. 13.

[180] Xudong Jia, Peng Xun, Wei Peng, Baokang Zhao, Haojie Li, and Chiran Shen. "Deep anomaly detection for time series: A survey." en. In: *Computer Science Review* 58 (Nov. 2025), p. 100787. ISSN: 15740137. DOI: 10.1016/j.cosrev.2025.100787. URL: https://linkinghub.elsevier.com/retrieve/pii/S1574013725000632 (visited on 09/22/2025).

[181] Maowei Jiang, Pengyu Zeng, Kai Wang, Huan Liu, Wenbo Chen, and Haoran Liu. "FECAM: Frequency Enhanced Channel Attention Mechanism for Time Series Forecasting." In: *Adv. Eng. Informatics* 58 (2022), p. 102158. DOI: 10.48550/arXiv.2212.01209.

[182] Kun Jin, Shaolong Sun, Hongtao Li, and Fengting Zhang. "A novel multi-modal analysis model with Baidu Search Index for subway passenger flow forecasting." In: *Engineering Applications of Artificial Intelligence* 107 (2022), p. 104518. URL: https://doi.org/10.1016/j.engappai.2021.104518.

[183] Ming Jin, Huan Yee Koh, Qingsong Wen, Daniele Zambon, Cesare Alippi, Geoffrey I Webb, Irwin King, and Shirui Pan. "A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[184] Ming Jin, Yu Zheng, Yuan-Fang Li, Siheng Chen, Bin Yang, and Shirui Pan. "Multivariate time series forecasting with dynamic graph neural odes." In: *IEEE Transactions on Knowledge and Data Engineering* 35.9 (2022), pp. 9168–9180.

[185] Michael I Jordan. "Serial order: A parallel distributed processing approach." In: *Advances in psychology*. Vol. 121. Elsevier, 1997, pp. 471–495.

[186] Satyadhar Joshi. "A Survey of Mixture of Experts Models: Architectures and Applications in Business and Finance." In: (2025). DOI: 10.2139/ssrn.5261449. URL: http://dx.doi.org/10.2139/ssrn.5261449.

[187] Rudolph Emil Kalman. "A new approach to linear filtering and prediction problems." In: (1960).

[188] Minseo Kang and Byunghan Lee. "Tictok: Time-series anomaly detection with contrastive tokenization." In: *IEEE Access* 11 (2023), pp. 81011–81020.

[189] Wang-Cheng Kang and Julian McAuley. "Self-Attentive Sequential Recommendation." In: *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, Nov. 2018, pp. 197–206.

DOI: 10.1109/icdm.2018.00035. URL: http://dx.doi.org/10.1109/icdm.2018.00035.

[190] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. "Scaling laws for neural language models." In: *arXiv preprint arXiv:2001.08361* (2020).

[191] Jacob Kauffmann, Klaus-Robert Müller, and Grégoire Montavon. "Towards explaining anomalies: a deep Taylor decomposition of one-class models." In: *Pattern Recognition* 101 (2020), p. 107198.

[192] Seyed Mehran Kazemi, Rishab Goel, Sepehr Eghbali, Janahan Ramanan, Jaspreet Sahota, Sanjay Thakur, Stella Wu, Cathal Smyth, Pascal Poupart, and Marcus Brubaker. "Time2vec: Learning a vector representation of time." In: *arXiv preprint arXiv:1907.05321* (2019). URL: https://doi.org/10.48550/arXiv.1907.05321.

[193] Jintao Ke, Hongyu Zheng, Hai Yang, and Xiqun Michael Chen. "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach." In: *Transportation research part C: Emerging technologies* 85 (2017), pp. 591–608. URL: https://doi.org/10.1016/j.trc.2017.10.016.

[194] Alec N. Kercheval and Yuan Zhang. "Modelling high-frequency limit order book dynamics with support vector machines." In: *Quantitative Finance* 15.8 (June 2015), pp. 1315–1329. ISSN: 1469-7696. DOI: 10.1080/14697688.2015.1032546. URL: http://dx.doi.org/10.1080/14697688.2015.1032546.

[195] Pouya Khodaee, Akbar Esfahanipour, and Hassan Mehtari Taheri. "Forecasting turning points in stock price by applying a novel hybrid CNN-LSTM-ResNet model fed by 2D segmented images." In: *Engineering Applications of Artificial Intelligence* 116 (2022), p. 105464. URL: https://doi.org/10.1016/j.engappai.2022.105464.

[196] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. "Supervised contrastive learning." In: *Advances in neural information processing systems* 33 (2020), pp. 18661–18673.

[197] *Kibana*. https://github.com/elastic/kibana.

[198] Tung Kieu, Bin Yang, Chenjuan Guo, and Christian S. Jensen. "Outlier Detection for Time Series with Recurrent Autoencoder Ensembles." In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*. IJCAI-2019. International Joint Conferences on Artificial Intelligence Organization, Aug. 2019, pp. 2725–2732. DOI: 10.24963/ijcai.2019/378. URL: http://dx.doi.org/10.24963/ijcai.2019/378.

[199] Ha Young Kim and Chang Hyun Won. "Forecasting the volatility of stock price index: A hybrid model integrating LSTM with multiple GARCH-type models." In: *Expert Systems with Applications* 103 (2018), pp. 25–37.

[200] Minha Kim, Kishor Kumar Bhaumik, Amin Ahsan Ali, and Simon S. Woo. "MIXAD: Memory-Induced Explainable Time Series Anomaly Detection." In: *Pattern Recognition*. Springer Nature Switzerland, Dec. 2024, pp. 242–257. ISBN: 9783031781896. DOI: 10.1007/978-3-031-78189-6_16. URL: http://dx.doi.org/10.1007/978-3-031-78189-6_16.

[201] DP Kingma and M Welling. "Auto-encoding variational Bayes. 2nd international conference on learning representations (ICLR2014)." In: *Preprint, submitted December* 23 (2014), p. 2013.

[202] Diederik P Kingma, Max Welling, et al. "An introduction to variational autoencoders." In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392.

[203] Polina Kirichenko, Pavel Izmailov, and Andrew G Wilson. "Why Normalizing Flows Fail to Detect Out-of-Distribution Data." In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 20578–20589. URL: https://proceedings.neurips.cc/paper_fil

es/paper/2020/file/ecb9fe2fbb99c31f567e9823e884dbec-Paper.pdf.

[204] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. "Reformer: The efficient transformer." In: *arXiv preprint arXiv:2001.04451* (2020). URL: https://doi.org/10.48550/arXiv.2001.04451.

[205] K A H Kobbacy, S Vadera, and M H Rasmy. "AI and OR in management of operations: history and trends." In: *Journal of the Operational Research Society* 58.1 (Jan. 2007), pp. 10–28. ISSN: 1476-9360. DOI: 10.1057/palgrave.jors.2602132. URL: http://dx.doi.org/10.1057/palgrave.jors.2602132.

[206] Vaia I Kontopoulou, Athanasios D Panagopoulos, Ioannis Kakkos, and George K Matsopoulos. "A review of ARIMA vs. machine learning approaches for time series forecasting in data driven networks." In: *Future Internet* 15.8 (2023), p. 255.

[207] Martin Kopp, Tomáš Pevnỳ, and Martin Holeňa. "Anomaly explanation with random forests." In: *Expert Systems with Applications* 149 (2020), p. 113187.

[208] Inès Ben Kraiem, Faiza Ghozzi, André Péninou, Geoffrey Roman-Jimenez, and Olivier Teste. "Human-interpretable rules for anomaly detection in time-series." In: *INTERNATIONAL CONFERENCE ON EXTENDING DATABASE TECHNOLOGY*. OpenProceedings. org. 2021, pp. 457–462.

[209] Hans-Peter Kriegel, Matthias Schubert, and Arthur Zimek. "Angle-based outlier detection in high-dimensional data." In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2008, pp. 444–452.

[210] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet classification with deep convolutional neural networks." In: *Communications of the ACM* 60.6 (May 2017), pp. 84–90. ISSN: 1557-7317. DOI: 10.1145/3065386. URL: http://dx.doi.org/10.1145/3065386.

[211] Subhadip Kumar. "Data Silos A Roadblock for AIOps." In: *arXiv preprint arXiv:2312.10039* (2023).

[212] Bum Chul Kwon, Min-Je Choi, Joanne Taery Kim, Edward Choi, Young Bin Kim, Soonwook Kwon, Jimeng Sun, and Jaegul Choo. "Retainvis: Visual analytics with interpretable and interactive recurrent neural networks on electronic medical records." In: *IEEE transactions on visualization and computer graphics* 25.1 (2018), pp. 299–309. URL: https://doi.org/10.1109/TVCG.2018.2865027.

[213] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. "Modeling long-and short-term temporal patterns with deep neural networks." In: *The 41st international ACM SIGIR conference on research & development in information retrieval*. 2018, pp. 95–104. URL: https://doi.org/10.48550/arXiv.1703.07015.

[214] Anukool Lakhina, Mark Crovella, and Christophe Diot. "Diagnosing network-wide traffic anomalies." In: *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*. SIGCOMM04. ACM, Aug. 2004, 219–230. DOI: 10.1145/1015467.1015492. URL: http://dx.doi.org/10.1145/1015467.1015492.

[215] Rocco Langone, Alfredo Cuzzocrea, and Nikolaos Skantzos. "Interpretable Anomaly Prediction: Predicting anomalous behavior in industry 4.0 settings via regularized logistic regression tools." In: *Data & Knowledge Engineering* 130 (2020), p. 101850.

[216] Nikolay Laptev, Saeed Amizadeh, and Ian Flint. "Generic and Scalable Framework for Automated Time-series Anomaly Detection." In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '15. ACM, Aug. 2015, 1939–1947. DOI: 10.1145/2783258.2788611. URL: http://dx.doi.org/10.1145/2783258.2788611.

[217] Pedro Lara-Benítez, Manuel Carranza-García, and José C Riquelme. "An experimental review on deep learning architectures for time series forecasting." In: *International journal of neural systems* 31.03 (2021), p. 2130001. URL: https://doi.org/10.1142/S0129065721300011.

[218] Alexander Lavin and Subutai Ahmad. "Evaluating real-time anomaly detection algorithms–the Numenta anomaly benchmark." In: *2015 IEEE 14th international conference on machine learning and applications (ICMLA)*. IEEE. 2015, pp. 38–44.

[219] Yann LeCun, Yoshua Bengio, et al. "Convolutional networks for images, speech, and time series." In: *The handbook of brain theory and neural networks* 3361.10 (1995), p. 1995.

[220] Hyunseong Lee, Guoyi Li, Ashwin Rai, and Aditi Chattopadhyay. "Real-time anomaly detection framework using a support vector regression for the safety monitoring of commercial aircraft." In: *Advanced Engineering Informatics* 44 (2020), p. 101071.

[221] Leonardo Leite, Carla Rocha, Fabio Kon, Dejan Milojicic, and Paulo Meirelles. "A survey of DevOps concepts and challenges." In: *ACM Computing Surveys (CSUR)* 52.6 (2019), pp. 1–35.

[222] José Leites, Vitor Cerqueira, and Carlos Soares. "Lag selection for univariate time series forecasting using deep learning: an empirical study." In: *EPIA Conference on Artificial Intelligence*. Springer. 2024, pp. 321–332.

[223] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. "Gshard: Scaling giant models with conditional computation and automatic sharding." In: *arXiv preprint arXiv:2006.16668* (2020).

[224] Andrew Lerner. *AIOps Platforms—Gartner*. URL: https://blogs.gartner.com/andrew-lerner/2017/08/09/aiops-platforms/.

[225] Marie-Jeanne Lesot and Adrien Revault d'Allonnes. "Credit-card fraud profiling using a hybrid incremental clustering methodology." In: *International Conference on Scalable Uncertainty Management*. Springer. 2012, pp. 325–336.

[226] Anna Levin, Shelly Garion, Elliot K Kolodner, Dean H Lorenz, Katherine Barabash, Mike Kugler, and Niall McShane. "AIOps for a cloud object storage service." In: *2019 IEEE International Congress on Big Data (BigDataCongress)*. IEEE. 2019, pp. 165–169.

[227] Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. "Base layers: Simplifying training of large, sparse models." In: *International Conference on Machine Learning*. PMLR. 2021, pp. 6265–6274.

[228] Edmond Lezmi and Jiali Xu. "Time Series Forecasting with Transformer Models and Application to Asset Management." In: *SSRN Electronic Journal* (2023). ISSN: 1556-5068. DOI: 10.2139/ssrn.4375798. URL: http://dx.doi.org/10.2139/ssrn.4375798.

[229] Bin Li and Emmanuel Müller. "Contrastive Time Series Anomaly Detection by Temporal Transformations." In: *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, June 2023, 1–8. DOI: 10.1109/ijcnn54540.2023.10191358. URL: http://dx.doi.org/10.1109/ijcnn54540.2023.10191358.

[230] Chuang Li, Zhecong Zhang, and Liping Wang. "Carbon peak forecast and low carbon policy choice of transportation industry in China: scenario prediction based on STIRPAT model." In: *Environmental Science and Pollution Research* 30.22 (2023), pp. 63250–63271. URL: http://dx.doi.org/10.1007/s11356-023-26549-6.

[231] Dan Li, Dacheng Chen, Jonathan Goh, and See-kiong Ng. "Anomaly detection with generative adversarial networks for multivariate time series." In: *arXiv preprint arXiv:1809.04758* (2018).

[232] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks." In: *International conference on artificial neural networks*. Springer. 2019, pp. 703–716.

[233] Gen Li and Jason J Jung. "Dynamic relationship identification for abnormality detection on financial time series." In: *Pattern Recognition Letters* 145 (2021), pp. 194–199.

[234] Gen Li and Jason J Jung. "Deep learning for anomaly detection in multivariate time series: Approaches, applications, and challenges." In: *Information Fusion* 91 (2023), pp. 93–102.

[235] Haoyuan Li and Yifan Li. "Anomaly detection methods based on GAN: a survey." In: *Applied Intelligence* 53.7 (2023), pp. 8209–8231.

[236] Jian Li, Pinjia He, Jieming Zhu, and Michael R Lyu. "Software defect prediction via convolutional neural network." In: *2017 IEEE international conference on software quality, reliability and security (QRS)*. IEEE. 2017, pp. 318–328.

[237] Qingliang Li, Huibowen Hao, Yang Zhao, Qingtian Geng, Guangjie Liu, Yu Zhang, and Fanhua Yu. "GANs-LSTM model for soil temperature estimation from meteorological: a new approach." In: *IEEE Access* 8 (2020), pp. 59427–59443. URL: https://doi.org/10.1109/ACCESS.2020.2982996.

[238] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting." In: *Advances in neural information processing systems* 32 (2019). URL: https://doi.org/10.48550/arXiv.1907.00235.

[239] Wenkai Li, Wenbo Hu, Ting Chen, Ning Chen, and Cheng Feng. "StackVAE-G: An efficient and interpretable model for time series anomaly detection." In: *AI Open* 3 (2022), pp. 101–110.

[240] Xiaolong Li, Gang Pan, Zhaohui Wu, Guande Qi, Shijian Li, Daqing Zhang, Wangsheng Zhang, and Zonghui Wang. "Prediction of urban human mobility using large-scale taxi traces and its applications." In: *Frontiers of computer science* 6 (2012), pp. 111–121. URL: http://dx.doi.org/10.1007/s11704-011-1192-6.

[241] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting." In: *arXiv preprint arXiv:1707.01926* (2017).

[242] Yangguang Li, Zhen Ming Jiang, Heng Li, Ahmed E Hassan, Cheng He, Ruirui Huang, Zhengda Zeng, Mian Wang, and Pinan Chen. "Predicting node failures in an ultra-large-scale cloud computing platform: an aiops solution." In: *ACM Transactions on Software Engineering and Methodology (TOSEM)* 29.2 (2020), pp. 1–24.

[243] Yifan Li, Xiaoyan Peng, Jia Zhang, Zhiyong Li, and Ming Wen. "DCT-GAN: dilated convolutional transformer-based GAN for time series anomaly detection." In: *IEEE Transactions on Knowledge and Data Engineering* 35.4 (2021), pp. 3632–3644.

[244] Youru Li, Zhenfeng Zhu, Deqiang Kong, Hua Han, and Yao Zhao. "EA-LSTM: Evolutionary attention-based LSTM for time series prediction." In: *Knowledge-Based Systems* 181 (2019), p. 104785. URL: https://doi.org/10.1016/j.knosys.2019.05.028.

[245] Zeyan Li, Wenxiao Chen, and Dan Pei. "Robust and unsupervised KPI anomaly detection based on conditional variational autoencoder." In: *2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC)*. IEEE. 2018, pp. 1–9.

[246] Zeyan Li, Chengyang Luo, Yiwei Zhao, Yongqian Sun, Kaixin Sui, Xiping Wang, Dapeng Liu, Xing Jin, Qi Wang, and Dan Pei. "Generic and robust localization of multi-dimensional root causes." In: *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE. 2019, pp. 47–57.

[247] Zhenchuan Li, Guanjun Liu, Shuo Wang, Shiyang Xuan, and Changjun Jiang. "Credit card fraud detection via kernel-based supervised hashing." In: *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE. 2018, pp. 1249–1254.

[248] Zhihan Li, Youjian Zhao, Jiaqi Han, Ya Su, Rui Jiao, Xidao Wen, and Dan Pei. "Multivariate Time Series Anomaly Detection and Interpretation using Hierarchical Inter-Metric and Temporal Embedding." In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. KDD '21. ACM, Aug. 2021, pp. 3220–3230. DOI: 10.1145/3447548.3467075. URL: http://dx.doi.org/10.1145/3447548.3467075.

[249] Zhong Li, Yuxuan Zhu, and Matthijs Van Leeuwen. "A Survey on Explainable Anomaly Detection." In: *ACM Transactions on Knowledge Discovery from Data* 18.1 (Sept. 2023), 1–54. ISSN: 1556-472X. DOI: 10.1145/3609333. URL: http://dx.doi.org/10.1145/3609333.

[250] Yinglung Liang, Yanyong Zhang, Hui Xiong, and Ramendra Sahoo. "Failure Prediction in IBM BlueGene/L Event Logs." In: *Seventh IEEE International Conference on Data Mining (ICDM 2007)*. IEEE, Oct. 2007. DOI: 10.1109/icdm.2007.46. URL: http://dx.doi.org/10.1109/icdm.2007.46.

[251] Lyuchao Liao, Yongqiang Wang, Fumin Zou, Shuoben Bi, Jinya Su, and Qi Sun. "A multi-sensory stimulating attention model for cities' taxi service demand prediction." In: *Scientific reports* 12.1 (2022), p. 3065. URL: https://doi.org/10.1038/s41598-022-07072-z.

[252] Shujian Liao, Hao Ni, Lukasz Szpruch, Magnus Wiese, Marc Sabate-Vidales, and Baoren Xiao. "Conditional sig-wasserstein gans for time series generation." In: *arXiv preprint arXiv:2006.05421* (2020).

[253] Andy Liaw, Matthew Wiener, et al. "Classification and regression by randomForest." In: *R news* 2.3 (2002), pp. 18–22.

[254] Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. "Temporal fusion transformers for interpretable multi-horizon time series forecasting." In: *International journal of forecasting* 37.4 (2021), pp. 1748–1764.

[255] Bryan Lim and Stefan Zohren. "Time-series forecasting with deep learning: a survey." In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*

379.2194 (Feb. 2021), p. 20200209. ISSN: 1471-2962. DOI: 10.1098/rsta.2020.0209. URL: http://dx.doi.org/10.1098/rsta.2020.0209.

[256] Swee Kiat Lim, Yi Loo, Ngoc-Trung Tran, Ngai-Man Cheung, Gemma Roig, and Yuval Elovici. "Doping: Generative data augmentation for unsupervised anomaly detection with gan." In: *2018 IEEE international conference on data mining (ICDM)*. IEEE. 2018, pp. 1122–1127.

[257] Fan Lin, Matt Beadon, Harish Dattatraya Dixit, Gautham Vunnam, Amol Desai, and Sriram Sankar. "Hardware remediation at scale." In: *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*. IEEE. 2018, pp. 14–17.

[258] Fred Lin, Keyur Muzumdar, Nikolay Pavlovich Laptev, Mihai-Valentin Curelea, Seunghak Lee, and Sriram Sankar. "Fast dimensional analysis for root cause investigation in a large-scale service environment." In: *Proceedings of the ACM on Measurement and Analysis of Computing Systems* 4.2 (2020), pp. 1–23.

[259] Zachary C Lipton. "The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery." In: *Queue* 16.3 (2018), pp. 31–57.

[260] Dapeng Liu, Youjian Zhao, Haowen Xu, Yongqian Sun, Dan Pei, Jiao Luo, Xiaowei Jing, and Mei Feng. "Opprentice: Towards Practical and Automatic Anomaly Detection Through Machine Learning." In: *Proceedings of the 2015 Internet Measurement Conference*. IMC '15. ACM, Oct. 2015, 211–224. DOI: 10.1145/2815675.2815679. URL: http://dx.doi.org/10.1145/2815675.2815679.

[261] Fan Liu, Xingshe Zhou, Jinli Cao, Zhu Wang, Tianben Wang, Hua Wang, and Yanchun Zhang. "Anomaly detection in quasi-periodic time series based on automatic data segmentation and attentional LSTM-CNN." In: *IEEE Transactions on Knowledge and Data Engineering* 34.6 (2020), pp. 2626–2640.

[262] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation Forest." In: *2008 Eighth IEEE International Conference on Data Mining*. IEEE, Dec. 2008, pp. 413–422. DOI: 10.1109/icdm.2008.17. URL: http://dx.doi.org/10.1109/icdm.2008.17.

[263] Jie Liu, Qilin Li, Senjian An, Bradley Ezard, and Ling Li. "EdgeConvFormer: Dynamic Graph CNN and Transformer based Anomaly Detection in Multivariate Time Series." In: *arXiv preprint arXiv:2312.01729* (2023).

[264] Lingbo Liu, Zhilin Qiu, Guanbin Li, Qing Wang, Wanli Ouyang, and Liang Lin. "Contextualized spatial–temporal network for taxi origin-destination demand prediction." In: *IEEE Transactions on Intelligent Transportation Systems* 20.10 (2019), pp. 3875–3887. URL: https://doi.org/10.1109/ICME.2019.00136.

[265] Qingxiang Liu, Chenghao Liu, Sheng Sun, Di Yao, and Yuxuan Liang. "GDformer: Going Beyond Subsequence Isolation for Multivariate Time Series Anomaly Detection." In: *arXiv preprint arXiv:2501.18196* (2025).

[266] Xu Liu, Juncheng Liu, Gerald Woo, Taha Aksu, Yuxuan Liang, Roger Zimmermann, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. "Moirai-moe: Empowering time series foundation models with sparse mixture of experts." In: *arXiv preprint arXiv:2410.10469* (2024).

[267] Yi Liu, Sahil Garg, Jiangtian Nie, Yang Zhang, Zehui Xiong, Jiawen Kang, and M Shamim Hossain. "Deep anomaly detection for time-series data in industrial IoT: A communication-efficient on-device federated learning approach." In: *IEEE Internet of Things Journal* 8.8 (2020), pp. 6348–6358.

[268] Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. "itransformer: Inverted transformers are effective for time series forecasting." In: *arXiv preprint arXiv:2310.06625* (2023).

[269] Zehao Liu, Mengzhou Gao, and Pengfei Jiao. "Gcad: Anomaly detection in multivariate time series from the perspective of

granger causality." In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 39. 18. 2025, pp. 19041–19049.

[270] Zhenyu Liu, Zhengtong Zhu, Jing Gao, and Cheng Xu. "Forecast methods for time series data: a survey." In: *Ieee Access* 9 (2021), pp. 91896–91912. URL: https://doi.org/10.1109/ACCESS.2021.3091162.

[271] Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y Hou, and Max Tegmark. "Kan: Kolmogorov-arnold networks." In: *arXiv preprint arXiv:2404.19756* (2024).

[272] *Logstash*. https://github.com/elastic/logstash.

[273] Thomas Louail, Maxime Lenormand, Oliva G Cantu Ros, Miguel Picornell, Ricardo Herranz, Enrique Frias-Martinez, José J Ramasco, and Marc Barthelemy. "From mobile phone data to the spatial structure of cities." In: *Scientific reports* 4.1 (2014), p. 5276. URL: https://doi.org/10.1038/srep05276.

[274] Scott Lundberg. "A unified approach to interpreting model predictions." In: *arXiv preprint arXiv:1705.07874* 30 (2017).

[275] *MLflow*. https://github.com/mlflow.

[276] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H. Chi. "Modeling Task Relationships in Multi-task Learning with Multi-gate Mixture-of-Experts." In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '18. ACM, July 2018, pp. 1930–1939. DOI: 10.1145/3219819.3220007. URL: http://dx.doi.org/10.1145/3219819.3220007.

[277] Yihong Ma, Md Nafee Al Islam, Jane Cleland-Huang, and Nitesh V. Chawla. "Detecting Anomalies in Small Unmanned Aerial Systems via Graphical Normalizing Flows." In: *IEEE Intelligent Systems* 38.2 (Mar. 2023), 46–54. ISSN: 1941-1294. DOI: 10.1109/mis.2023.3252810. URL: http://dx.doi.org/10.1109/mis.2023.3252810.

[278] Youzhong Ma, Jia Rao, Weisong Hu, Xiaofeng Meng, Xu Han, Yu Zhang, Yunpeng Chai, and Chunqiu Liu. "An efficient index for massive IOT data in cloud environment." In: *Proceedings of the 21st ACM international conference on Information and knowledge management*. 2012, pp. 2129–2133.

[279] Meghanath Macha and Leman Akoglu. "Explaining anomalies in groups with characterizing subspace rules." In: *Data Mining and Knowledge Discovery* 32 (2018), pp. 1444–1480.

[280] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. "Statistical and Machine Learning forecasting methods: Concerns and ways forward." In: *PloS one* 13.3 (2018), e0194889. URL: https://doi.org/10.1371/journal.pone.0194889.

[281] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. "The M4 Competition: Results, findings, conclusion and way forward." In: *International Journal of forecasting* 34.4 (2018), pp. 802–808.

[282] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. "The M4 Competition: 100,000 time series and 61 forecasting methods." In: *International Journal of Forecasting* 36.1 (2020), pp. 54–74.

[283] Spyros Makridakis, Evangelos Spiliotis, Vassilios Assimakopoulos, Artemios-Anargyros Semenoglou, Gary Mulder, and Konstantinos Nikolopoulos. "Statistical, machine learning and deep learning forecasting methods: Comparisons and ways forward." In: *Journal of the Operational Research Society* 74.3 (2023), pp. 840–859.

[284] Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, Puneet Agarwal, et al. "Long short term memory networks for anomaly detection in time series." In: *Esann*. Vol. 2015. 2015, p. 89.

[285] Massimiliano Marcellino, James H Stock, and Mark W Watson. "A comparison of direct and iterated multistep AR methods for forecasting macroeconomic time series." In: *Journal of econometrics* 135.1-2 (2006), pp. 499–526.

[286] Ioulia Markou, Filipe Rodrigues, and Francisco C Pereira. "Use of taxi-trip data in analysis of demand patterns for detection and explanation of anomalies." In: *Transportation Research Record* 2643.1 (2017), pp. 129–138.

[287] Chihiro Maru and Ichiro Kobayashi. "Collective anomaly detection for multivariate data using generative adversarial networks." In: *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE. 2020, pp. 598–604.

[288] Adnan Masood, Adnan Hashmi, Adnan Masood, and Adnan Hashmi. "AIOps: predictive analytics & machine learning in operations." In: *Cognitive computing recipes: Artificial intelligence solutions using microsoft cognitive services and TensorFlow* (2019), pp. 359–382.

[289] Warren S. McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." In: *The Bulletin of Mathematical Biophysics* 5.4 (Dec. 1943), pp. 115–133. ISSN: 1522-9602. DOI: 10.1007/bf02478259. URL: http://dx.doi.org/10.1007/bf02478259.

[290] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality." In: *Advances in neural information processing systems* 26 (2013). URL: https://doi.org/10.48550/arXiv.1310.4546.

[291] Tim Miller, Piers Howe, and Liz Sonenberg. "Explainable AI: Beware of inmates running the asylum or: How I learnt to stop worrying and love the social and behavioural sciences." In: *arXiv preprint arXiv:1712.00547* (2017).

[292] Olof Mogren. "C-RNN-GAN: Continuous recurrent neural networks with adversarial training." In: *arXiv preprint arXiv:1611.09904* (2016).

[293] Luis Moreira-Matias, Joao Gama, Michel Ferreira, Joao Mendes-Moreira, and Luis Damas. "Predicting taxi–passenger demand using streaming data." In: *IEEE Transactions on*

Intelligent Transportation Systems 14.3 (2013), pp. 1393–1402. URL: https://doi.org/10.1109/TITS.2013.2262376.

[294] Manfred Mudelsee. "Trend analysis of climate time series: A review of methods." In: *Earth-science reviews* 190 (2019), pp. 310–322.

[295] Mukosi Abraham Mukwevho and Turgay Celik. "Toward a Smart Cloud: A Review of Fault-Tolerance Methods in Cloud Systems." In: *IEEE Transactions on Services Computing* 14.2 (Mar. 2021), pp. 589–605. ISSN: 2372-0204. DOI: 10.1109/tsc.2018.2816644. URL: http://dx.doi.org/10.1109/tsc.2018.2816644.

[296] Pavol Mulinka, Pedro Casas, Kensuke Fukuda, and Lukas Kencl. "HUMAN-Hierarchical Clustering for Unsupervised Anomaly Detection & Interpretation." In: *2020 11th International Conference on Network of the Future (NoF)*. IEEE. 2020, pp. 132–140.

[297] 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. ISSN: 2169-3536. DOI: 10.1109/access.2018.2886457. URL: http://dx.doi.org/10.1109/access.2018.2886457.

[298] Mohsin Munir, Shoaib Ahmed Siddiqui, Ferdinand Küsters, Dominique Mercier, Andreas Dengel, and Sheraz Ahmed. "Tsxplain: Demystification of dnn decisions for time-series using natural language and statistical features." In: *International conference on artificial neural networks*. Springer. 2019, pp. 426–439.

[299] N Muraleedharan and B Janet. "A deep learning based HTTP slow DoS classification approach using flow data." In: *ICT Express* 7.2 (2021), pp. 210–214.

[300] Małgorzata Murat, Iwona Malinowska, Magdalena Gos, and Jaromir Krzyszczak. "Forecasting daily meteorological time series using ARIMA and regression models." In: *International Agrophysics* 32.2 (Apr. 2018), pp. 253–264. ISSN: 2300-8725.

DOI: 10.1515/intag-2017-0007. URL: http://dx.doi.org/10.1515/intag-2017-0007.

[301] Joseph F Murray, Gordon F Hughes, Kenneth Kreutz-Delgado, and Dale Schuurmans. "Machine Learning Methods for Predicting Failures in Hard Drives: A Multiple-Instance Application." In: *Journal of Machine Learning Research* 6.5 (2005).

[302] Susumu Naito, Yasunori Taguchi, Kouta Nakata, and Yuichi Kato. "Anomaly Detection for Multivariate Time Series on Large-scale Fluid Handling Plant Using Two-stage Autoencoder." In: *2021 International Conference on Data Mining Workshops (ICDMW)*. IEEE, Dec. 2021, pp. 542–551. DOI: 10.1109/icdmw53433.2021.00072. URL: http://dx.doi.org/10.1109/icdmw53433.2021.00072.

[303] Masashi Nakajima. *Payment system technologies and functions: innovations and developments*. IGI Global, 2011.

[304] Takuto Nakashima and Takehisa Yairi. "Assessing the Performance of Transformer for Time Series Anomaly Detection." In: *PHM Society Asia-Pacific Conference* 4.1 (Sept. 2023). ISSN: 2994-7219. DOI: 10.36001/phmap.2023.v4i1.3773. URL: http://dx.doi.org/10.36001/phmap.2023.v4i1.3773.

[305] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, and Balaji Lakshminarayanan. "Detecting out-of-distribution inputs to deep generative models using typicality." In: *arXiv preprint arXiv:1906.02994* (2019).

[306] Sasho Nedelkoski, Jorge Cardoso, and Odej Kao. "Anomaly detection and classification using distributed tracing and deep learning." In: *2019 19th IEEE/ACM international symposium on cluster, cloud and grid computing (CCGRID)*. IEEE. 2019, pp. 241–250.

[307] Oliver Neumann, Maximilian Beichter, Benedikt Heidrich, Nils Friederich, Veit Hagenmeyer, and Ralf Mikut. "Intrinsic Explainable Artificial Intelligence Using Trainable Spatial Weights on Numerical Weather Predictions." In: *The 15th ACM International Conference on Future and Sustainable Energy Systems*. e-Energy '24. ACM, May 2024, pp. 551–559. DOI:

10.1145/3632775.3662161. URL: http://dx.doi.org/10.1145/3632775.3662161.

[308] Quoc Phong Nguyen, Kar Wai Lim, Dinil Mon Divakaran, Kian Hsiang Low, and Mun Choon Chan. "Gee: A gradient-based explainable variational autoencoder for network anomaly detection." In: *2019 IEEE Conference on Communications and Network Security (CNS)*. IEEE. 2019, pp. 91–99.

[309] Zijian Niu, Ke Yu, and Xiaofei Wu. "LSTM-based VAE-GAN for time-series anomaly detection." In: *Sensors* 20.13 (2020), p. 3738.

[310] Paolo Notaro, Jorge Cardoso, and Michael Gerndt. "A systematic mapping study in AIOps." In: *International Conference on Service-Oriented Computing*. Springer. 2020, pp. 110–123.

[311] Paolo Notaro, Jorge Cardoso, and Michael Gerndt. "A survey of aiops methods for failure management." In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 12.6 (2021), pp. 1–45.

[312] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. "Wavenet: A generative model for raw audio." In: *arXiv preprint arXiv:1609.03499* (2016).

[313] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. "N-BEATS: Neural basis expansion analysis for interpretable time series forecasting." In: *arXiv preprint arXiv:1905.10437* (2019). URL: https://doi.org/10.48550/arXiv.1905.10437.

[314] Dheeraj Kumar Dukhiram Pal, Venkat Rama Raju Alluri, Shashi Thota, Venkata Sri Manoj Bonam, Subrahmanyasarma Chitta, and Mahammad Shaik. "AIOps: Integrating AI and Machine Learning into IT Operations." In: *Australian Journal of Machine Learning Research & Applications* 4.1 (2024), pp. 288–311.

[315] Guansong Pang and Charu Aggarwal. "Toward explainable deep anomaly detection." In: *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2021, pp. 4056–4057.

[316] John Paparrizos, Paul Boniol, Themis Palpanas, Ruey S. Tsay, Aaron Elmore, and Michael J. Franklin. "Volume under the surface: a new accuracy evaluation measure for time-series anomaly detection." In: *Proceedings of the VLDB Endowment* 15.11 (July 2022), pp. 2774–2787. ISSN: 2150-8097. DOI: 10.14778/3551793.3551830. URL: http://dx.doi.org/10.14778/3551793.3551830.

[317] Daehyung Park, Yuuna Hoshi, and Charles C Kemp. "A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder." In: *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 1544–1551.

[318] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. "Pytorch: An imperative style, high-performance deep learning library." In: *Advances in neural information processing systems* 32 (2019).

[319] Jayesh Patel. "Bridging data silos using big data integration." In: *International Journal of Database Management Systems* 11.3 (2019), pp. 01–06.

[320] Jeffrey Pennington, Richard Socher, and Christopher D Manning. "Glove: Global vectors for word representation." In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543. URL: http://dx.doi.org/10.3115/v1/D14-1162.

[321] Omkar Reddy Polu. "AI-driven prognostic failure analysis for autonomous resilience in cloud data centers." In: *International Journal of Cloud Computing* 2.2 (Dec. 2024), pp. 27–37. DOI: 10.34218/ijcc_02_02_003. URL: http://dx.doi.org/10.34218/ijcc\%5F02\%5F02\%5F003.

[322] Lorenzo Porcelli. "Business-Aware Failure Detection in Instant Payment Infrastructures." In: *arXiv preprint arXiv:2510.21710* (2025). DOI: 10.48550/arXiv.2510.21710. URL: https://arxiv.org/abs/2510.21710.

[323] Lorenzo Porcelli. *The Evolution of IT Operations in TIPS: From Monitoring to AI-Driven Anomaly Detection*. Research and analysis. Submitted for publication in Markets, Infrastructures, Payment Systems. Bank of Italy, 2025.

[324] Lorenzo Porcelli, Ugo Fiore, and Francesco Palmieri. "Generative models with helical time encoding for seasonal time series forecasting." In: *Engineering Applications of Artificial Intelligence* 162 (Dec. 2025), p. 112780. ISSN: 0952-1976. DOI: 10.1016/j.engappai.2025.112780. URL: http://dx.doi.org/10.1016/j.engappai.2025.112780.

[325] Lorenzo Porcelli, Marcello Trovati, and Francesco Palmieri. "Real-time anomaly detection in seasonal time series with conditional variational autoencoder." In: *Applied Soft Computing* 184 (Dec. 2025), p. 113761. ISSN: 1568-4946. DOI: 10.1016/j.asoc.2025.113761. URL: http://dx.doi.org/10.1016/j.asoc.2025.113761.

[326] Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison W. Cottrell. "A Dual-Stage Attention-Based Recurrent Neural Network for Time Series Prediction." In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*. IJCAI-2017. International Joint Conferences on Artificial Intelligence Organization, Aug. 2017, pp. 2627–2633. DOI: 10.24963/ijcai.2017/366. URL: http://dx.doi.org/10.24963/ijcai.2017/366.

[327] Zhilin Qiu, Lingbo Liu, Guanbin Li, Qing Wang, Nong Xiao, and Liang Lin. "Taxi origin-destination demand prediction with contextualized spatial-temporal network." In: *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE. 2019, pp. 760–765. URL: https://doi.org/10.1109/ICME.2019.00136.

[328] Mohammad Hossein Rafiei and Hojjat Adeli. "A novel machine learning model for estimation of sale prices of real estate units." In: *Journal of Construction Engineering and Management* 142.2 (2016), p. 04015066.

[329] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. "Zero-shot text-to-image generation." In: *International conference on machine learning*. Pmlr. 2021, pp. 8821–8831.

[330] Erica Ramirez, Markus Wimmer, and Martin Atzmueller. "A computational framework for interpretable anomaly detection and classification of multivariate time series with application to human gait data analysis." In: *Artificial Intelligence in Medicine: Knowledge Representation and Transparent and Explainable Systems: AIME 2019 International Workshops, KR4HC/ProHealth and TEAAM, Poznan, Poland, June 26–29, 2019, Revised Selected Papers*. Springer. 2019, pp. 132–147.

[331] Syama Sundar Rangapuram, Matthias W Seeger, Jan Gasthaus, Lorenzo Stella, Yuyang Wang, and Tim Januschowski. "Deep state space models for time series forecasting." In: *Advances in neural information processing systems* 31 (2018). URL: https://proceedings.neurips.cc/paper_files/paper/2018/file/5cf68969fb67aa6082363a6d4e6468e2-Paper.pdf.

[332] Chotirat Ann Ratanamahatana, Jessica Lin, Dimitrios Gunopulos, Eamonn Keogh, Michail Vlachos, and Gautam Das. "Mining time series data." In: *Data mining and knowledge discovery handbook*. Springer, 2010, pp. 1049–1077.

[333] Mohammad Abdur Razzaque, Marija Milojevic-Jevric, Andrei Palade, and Siobhán Clarke. "Middleware for internet of things: a survey." In: *IEEE Internet of things journal* 3.1 (2015), pp. 70–95.

[334] Ye Ren, P.N. Suganthan, and N. Srikanth. "Ensemble methods for wind and solar power forecasting—A state-of-the-art review." In: *Renewable and Sustainable Energy Reviews* 50 (Oct. 2015), pp. 82–91. ISSN: 1364-0321. DOI: 10.1016/j.rser.2015.

04.081. URL: http://dx.doi.org/10.1016/j.rser.2015.04.081.

[335] Massimiliano Renzetti, Serena Bernardini, Giuseppe Marino, Luca Mibelli, Laura Ricciardi, and Giovanni Maria Sabelli. "TIPS - TARGET Instant Payment Settlement The Pan-European Infrastructure for the Settlement of Instant Payments." In: *Markets, Infrastructures, Payment Systems* 1 (2021). URL: https://www.bancaditalia.it/pubblicazioni/mercati-infrastrutture-e-sistemi-di-pagamento/questioni-istituzionali/2021-001/index.html.

[336] Massimiliano Renzetti, Fabrizio Dinacci, and Ann Börestam. "Cross-Currency Settlement of Instant Payments in a Multi-Currency Clearing and Settlement Mechanism." In: *Markets, Infrastructures, Payment Systems* 16 (2022). URL: https://www.bancaditalia.it/pubblicazioni/mercati-infrastrutture-e-sistemi-di-pagamento/approfondimenti/2022-028/index.html.

[337] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why should i trust you?" Explaining the predictions of any classifier." In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. 2016, pp. 1135–1144.

[338] Rodney Rick and Lilian Berton. "Energy forecasting model based on CNN-LSTM-AE for many time series with unequal lengths." In: *Engineering Applications of Artificial Intelligence* 113 (2022), p. 104998. URL: https://doi.org/10.1016/j.engappai.2022.104998.

[339] S. W. Roberts. "Control Chart Tests Based on Geometric Moving Averages." In: *Technometrics* 42.1 (Feb. 2000), 97–101. ISSN: 1537-2723. DOI: 10.1080/00401706.2000.10485986. URL: http://dx.doi.org/10.1080/00401706.2000.10485986.

[340] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. "Temporal graph networks for deep learning on dynamic graphs." In: *arXiv preprint arXiv:2006.10637* (2020).

[341] Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*. Inc., New York: John wiley & sons, 1987.

[342] Lukas Ruff, Jacob R. Kauffmann, Robert A. Vandermeulen, Gregoire Montavon, Wojciech Samek, Marius Kloft, Thomas G. Dietterich, and Klaus-Robert Muller. "A Unifying Review of Deep and Shallow Anomaly Detection." In: *Proceedings of the IEEE* 109.5 (May 2021), 756–795. ISSN: 1558-2256. DOI: 10.1109/jproc.2021.3052449. URL: http://dx.doi.org/10.1109/jproc.2021.3052449.

[343] T Konstantin Rusch and Siddhartha Mishra. "Unicornn: A recurrent model for learning very long time dependencies." In: *International Conference on Machine Learning*. PMLR. 2021, pp. 9168–9178.

[344] Waddah Saeed and Christian Omlin. "Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities." In: *Knowledge-Based Systems* 263 (2023), p. 110273.

[345] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. "Photorealistic text-to-image diffusion models with deep language understanding." In: *Advances in neural information processing systems* 35 (2022), pp. 36479–36494.

[346] Felix Salfner and Miroslaw Malek. "Using Hidden Semi-Markov Models for Effective Online Failure Prediction." In: *2007 26th IEEE International Symposium on Reliable Distributed Systems (SRDS 2007)*. IEEE, Oct. 2007. DOI: 10.1109/srds.2007.35. URL: http://dx.doi.org/10.1109/srds.2007.35.

[347] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. "Improved techniques for training gans." In: *Advances in neural information processing systems* 29 (2016).

[348] David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. "High-dimensional multivariate forecasting with low-rank gaussian copula processes." In: *Advances in neural information processing systems* 32 (2019).

[349] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. "DeepAR: Probabilistic forecasting with autoregressive recurrent networks." In: *International journal of forecasting* 36.3 (2020), pp. 1181–1191. URL: https://doi.org/10.1016/j.ijforecast.2019.07.001.

[350] Simo Särkkä and Lennart Svensson. *Bayesian filtering and smoothing*. Vol. 17. Cambridge university press, 2023.

[351] Anton Maximilian Schäfer and Hans Georg Zimmermann. "Recurrent neural networks are universal approximators." In: *International conference on artificial neural networks*. Springer. 2006, pp. 632–640.

[352] Thomas Schlegl, Stefan Schlegl, Nikolai West, and Jochen Deuse. "Scalable anomaly detection in manufacturing systems using an interpretable deep learning approach." In: *Procedia CIRP* 104 (2021), pp. 1547–1552.

[353] Thomas Schlegl, Philipp Seeböck, Sebastian M Waldstein, Ursula Schmidt-Erfurth, and Georg Langs. "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery." In: *International conference on information processing in medical imaging*. Springer. 2017, pp. 146–157.

[354] Jürgen Schmidhuber. "Deep learning in neural networks: An overview." In: *Neural networks* 61 (2015), pp. 85–117.

[355] Bernhard Schölkopf, Robert C Williamson, Alex Smola, John Shawe-Taylor, and John Platt. "Support vector method for novelty detection." In: *Advances in neural information processing systems* 12 (1999).

[356] B Schwartz and P Jinka. *Anomaly Detection for Monitoring*. 2015.

[357] Rajat Sen, Hsiang-Fu Yu, and Inderjit S Dhillon. "Think globally, act locally: A deep neural network approach to high-dimensional time series forecasting." In: *Advances in neural information processing systems* 32 (2019).

[358] Arnaldo Sgueglia, Andrea Di Sorbo, Corrado Aaron Visaggio, and Gerardo Canfora. "A systematic literature review of IoT time series anomaly detection solutions." en. In: *Future Generation Computer Systems* 134 (Sept. 2022), pp. 170–186. ISSN: 0167739X. DOI: 10.1016/j.future.2022.04.005. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167739X22001285 (visited on 09/22/2025).

[359] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. "Outrageously large neural networks: The sparsely-gated mixture-of-experts layer." In: *arXiv preprint arXiv:1701.06538* (2017).

[360] Lifeng Shen, Zhongzhong Yu, Qianli Ma, and James T. Kwok. "Time Series Anomaly Detection with Multiresolution Ensemble Decoding." In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.11 (May 2021), pp. 9567–9575. ISSN: 2159-5399. DOI: 10.1609/aaai.v35i11.17152. URL: http://dx.doi.org/10.1609/aaai.v35i11.17152.

[361] Manish Shetty et al. "Building AI Agents for Autonomous Clouds: Challenges and Design Principles." In: *Proceedings of the ACM Symposium on Cloud Computing*. SoCC '24. ACM, Nov. 2024, pp. 99–110. DOI: 10.1145/3698038.3698525. URL: http://dx.doi.org/10.1145/3698038.3698525.

[362] Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. "Time-moe: Billion-scale time series foundation models with mixture of experts." In: *arXiv preprint arXiv:2409.16040* (2024).

[363] Shun-Yao Shih, Fan-Keng Sun, and Hung-yi Lee. "Temporal pattern attention for multivariate time series forecasting." In: *Machine Learning* 108.8–9 (June 2019), pp. 1421–1441. ISSN: 1573-0565. DOI: 10.1007/s10994-019-05815-0. URL: http://dx.doi.org/10.1007/s10994-019-05815-0.

[364] Hamed Shirzad, Honghao Lin, Balaji Venkatachalam, Ameya Velingker, David P Woodruff, and Danica J Sutherland. "Even sparser graph transformers." In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 71277–71305.

[365] Amit K Shukla, Grégory Smits, Olivier Pivert, and Marie-Jeanne Lesot. "Explaining data regularities and anomalies." In: *2020 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*. IEEE. 2020, pp. 1–8.

[366] Haotian Si, Changhua Pei, Zhihan Li, Yadong Zhao, Jingjing Li, Haiming Zhang, Zulong Diao, Jianhui Li, Gaogang Xie, and Dan Pei. "Beyond sharing: Conflict-aware multivariate time series anomaly detection." In: *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 2023, pp. 1635–1645.

[367] Haotian Si et al. "TimeSeriesBench: An Industrial-Grade Benchmark for Time Series Anomaly Detection Models." In: *2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, Oct. 2024, pp. 61–72. DOI: 10.1109/issre62328.2024.00017. URL: http://dx.doi.org/10.1109/issre62328.2024.00017.

[368] Md Amran Siddiqui, Alan Fern, Thomas G Dietterich, and Weng-Keen Wong. "Sequential feature explanations for anomaly detection." In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 13.1 (2019), pp. 1–22.

[369] Alban Siffer, Pierre-Alain Fouque, Alexandre Termier, and Christine Largouet. "Anomaly detection in streams with extreme value theory." In: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM. 2017, pp. 1067–1075.

[370] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. "Mastering the game of Go with deep neural networks and tree search." In: *nature* 529.7587 (2016), pp. 484–489.

[371] Joakim Skarding, Bogdan Gabrys, and Katarzyna Musial. "Foundations and Modeling of Dynamic Networks Using Dynamic Graph Neural Networks: A Survey." In: *IEEE Access* 9 (2021), pp. 79143–79168. ISSN: 2169-3536. DOI: 10.1109/

access.2021.3082932. URL: http://dx.doi.org/10.1109/access.2021.3082932.

[372] *Skyline*. https://github.com/etsy/skyline. Accessed: 2024-07-10. 2013.

[373] Giulia Slavic, Pablo Marin, David Martin, Lucio Marcenaro, and Carlo Regazzoni. "Interpretable Anomaly Detection Using A Generalized Markov Jump Particle Filter." In: *2021 IEEE International Conference on Autonomous Systems (ICAS)*. IEEE. 2021, pp. 1–5.

[374] Alison Smith-Renner, Rob Rua, and Mike Colony. "Towards an Explainable Threat Detection Tool." In: *IUI workshops*. 2019.

[375] Slawek Smyl. "A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting." In: *International journal of forecasting* 36.1 (2020), pp. 75–85.

[376] Fei Song, Yanlei Diao, Jesse Read, Arnaud Stiegler, and Albert Bifet. "EXAD: A system for explainable anomaly detection on big data traces." In: *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE. 2018, pp. 1435–1440.

[377] Junho Song, Keonwoo Kim, Jeonglyul Oh, and Sungzoon Cho. "Memto: Memory-guided transformer for multivariate time series anomaly detection." In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 57947–57963.

[378] Pengyu Song, Chunhui Zhao, Biao Huang, and Jinliang Ding. "Explicit Representation and Customized Fault Isolation Framework for Learning Temporal and Spatial Dependencies in Industrial Processes." In: *IEEE Transactions on Neural Networks and Learning Systems* 35.3 (Mar. 2024), pp. 2997–3011. ISSN: 2162-2388. DOI: 10.1109/tnnls.2023.3262277. URL: http://dx.doi.org/10.1109/tnnls.2023.3262277.

[379] Xiaobao Song, Liwei Deng, Hao Wang, Yaoan Zhang, Yuxin He, and Wenming Cao. "Deep learning-based time series forecasting." In: *Artificial Intelligence Review* 58.1 (2024), p. 23.

[380] Xiuyao Song, Mingxi Wu, Christopher Jermaine, and Sanjay Ranka. "Conditional anomaly detection." In: *IEEE Transactions on knowledge and Data Engineering* 19.5 (2007), pp. 631–645.

[381] Sandeep Kumar Sood et al. "A scientometric analysis of quantum driven innovations in intelligent transportation systems." In: *Engineering Applications of Artificial Intelligence* 138 (2024), p. 109258. URL: https://doi.org/10.1016/j.engappai.2024.109258.

[382] 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* 38.3 (2024), pp. 1027–1068.

[383] Ryan E Sperl and Soon M Chung. "Two-step anomaly detection for time series data." In: *2019 International Conference on Data and Software Engineering (ICoDSE)*. IEEE. 2019, pp. 1–5.

[384] Philip Stahmann, Maximilian Nebel, and Christian Janiesch. "AI-based real-time anomaly detection in industrial engineering: A structured literature review, taxonomy, and research agenda." en. In: *Computers & Industrial Engineering* 207 (Sept. 2025), p. 111236. ISSN: 03608352. DOI: 10.1016/j.cie.2025.111236. URL: https://linkinghub.elsevier.com/retrieve/pii/S0360835225003821 (visited on 09/22/2025).

[385] Stephen V. Stehman. "Selecting and interpreting measures of thematic classification accuracy." In: *Remote Sensing of Environment* 62.1 (Oct. 1997), pp. 77–89. ISSN: 0034-4257. DOI: 10.1016/s0034-4257(97)00083-7. URL: http://dx.doi.org/10.1016/s0034-4257(97)00083-7.

[386] 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 & data mining*. 2019, pp. 2828–2837.

[387] He Sun, Zhun Deng, Hui Chen, and David Parkes. "Decision-aware conditional gans for time series data." In: *Proceedings of*

the Fourth ACM International Conference on AI in Finance. 2023, pp. 36–45.

[388] Yuting Sun, Guansong Pang, Guanhua Ye, Tong Chen, Xia Hu, and Hongzhi Yin. "Unraveling the 'anomaly'in time series anomaly detection: A self-supervised tri-domain solution." In: *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE. 2024, pp. 981–994.

[389] I Sutskever. "Sequence to Sequence Learning with Neural Networks." In: *arXiv preprint arXiv:1409.3215* (2014). URL: https://doi.org/10.48550/arXiv.1409.3215.

[390] THUML. *Autoformer*. GitHub. 2021. URL: https://github.com/thuml/Autoformer.

[391] Souhaib Ben Taieb, Antti Sorjamaa, and Gianluca Bontempi. "Multiple-output modeling for multi-step-ahead time series forecasting." In: *Neurocomputing* 73.10-12 (2010), pp. 1950–1957.

[392] Liaoyuan Tang, Zheng Wang, Guanxiong He, Rong Wang, and Feiping Nie. "Perturbation guiding contrastive representation learning for time series anomaly detection." In: *Proc. 33rd Int. Joint Conf. Artif. Intell*. 2024, pp. 4955–4963.

[393] Yuqing Tang, Fusheng Yu, W. Pedrycz, Xiyang Yang, Jiayin Wang, and Shihu Liu. "Building Trend Fuzzy Granulation-Based LSTM Recurrent Neural Network for Long-Term Time-Series Forecasting." In: *IEEE Transactions on Fuzzy Systems* 30 (2022), pp. 1599–1613. DOI: 10.1109/TFUZZ.2021.3062723.

[394] L Tarassenko, P Hayton, N Cerneaz, and M Brady. "Novelty detection for the identification of masses in mammograms." In: *1995 Fourth International Conference on Artificial Neural Networks*. IET. 1995, pp. 442–447.

[395] Nesime Tatbul, Taejun Lee, Stan Zdonik, Murali Alam, and Justin Gottschlich. "Precision and recall for time series." In: *Advances in Neural Information Processing Systems*. Vol. 31. 2018.

[396] Liangyu Tay, Joanne Mun-Yee Lim, Shiuan-Ni Liang, Chua Kah Keong, and Yong Haur Tay. "Urban traffic volume estimation using intelligent transportation system crowdsourced data." In: *Engineering Applications of Artificial Intelligence* 126 (2023), p. 107064. URL: https://doi.org/10.1016/j.engappai.2023.107064.

[397] Sean J Taylor and Benjamin Letham. "Forecasting at scale." In: *The American Statistician* 72.1 (2018), pp. 37–45.

[398] Hwee San. Teng, K. Chen, and Shuxia Lu. "Adaptive real-time anomaly detection using inductively generated sequential patterns." In: *Proceedings. 1990 IEEE Computer Society Symposium on Research in Security and Privacy* (1990), pp. 278–284.

[399] *TensorBoard*. https://github.com/tensorflow/tensorboard.

[400] *TensorFlow*. https://github.com/tensorflow/tensorflow.

[401] Hoang Thanh-Tung, Truyen Tran, and Svetha Venkatesh. "Improving generalization and stability of generative adversarial networks." In: *arXiv preprint arXiv:1902.03984* (2019).

[402] The SWIFT Standards Team. *ISO 20022 for dummies*. John Wiley & Sons, Ltd, 2022.

[403] Yongxin Tong, Yuqiang Chen, Zimu Zhou, Lei Chen, Jie Wang, Qiang Yang, Jieping Ye, and Weifeng Lv. "The simpler the better: a unified approach to predicting original taxi demands based on large-scale online platforms." In: *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2017, pp. 1653–1662. URL: http://dx.doi.org/10.1145/3097983.3098018.

[404] Eric J Topol. "High-performance medicine: the convergence of human and artificial intelligence." In: *Nature medicine* 25.1 (2019), pp. 44–56.

[405] Franca Rocco di Torrepadula, Enea Vincenzo Napolitano, Sergio Di Martino, and Nicola Mazzocca. "Machine Learning for public transportation demand prediction: A Systematic Literature Review." In: *Engineering Applications of Artificial*

Intelligence 137 (2024), p. 109166. URL: https://doi.org/10.1016/j.engappai.2024.109166.

[406] Mark Treveil, Nicolas Omont, Clément Stenac, Kenji Lefevre, Du Phan, Joachim Zentici, Adrien Lavoillotte, Makoto Miyazaki, and Lynn Heidmann. *Introducing MLOps*. O'Reilly Media, 2020.

[407] Anton Tsitsulin, John Palowitch, Bryan Perozzi, and Emmanuel Müller. "Graph clustering with graph neural networks." In: *Journal of Machine Learning Research* 24.127 (2023), pp. 1–21.

[408] 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.

[409] United States Centers for Disease Control and Prevention. *National, Regional, and State Level Outpatient Illness and Viral Surveillance*. https://gis.cdc.gov/grasp/fluview/fluportaldashboard.html. Accessed: 2025-09-01.

[410] Latha Narayanan Valli, N Sujatha, and E Joice Rathinam. "A Study on Deep Learning Frameworks to Understand the Real Time Fault Detection and Diagnosis in IT Operations with AIOPs." In: *2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT)*. IEEE. 2023, pp. 1–6.

[411] Owen Vallis, Jordan Hochenbaum, and Arun Kejariwal. "A novel technique for {Long-Term} anomaly detection in the cloud." In: *6th USENIX workshop on hot topics in cloud computing (HotCloud 14)*. 2014.

[412] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In: *Advances in neural information processing systems* 30 (2017). URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[413] Volt. *Real-time payments world map*. 2024. URL: https://www.volt.io/real-time-payments-world-map/ (visited on 10/29/2024).

[414] Laura Von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, et al. "Informed Machine Learning–A Taxonomy and Survey of Integrating Knowledge into Learning Systems." In: *arXiv preprint arXiv:1903.12394* (2019).

[415] Dennis Wagner, Tobias Michels, Florian CF Schulz, Arjun Nair, Maja Rudolph, and Marius Kloft. "Timesead: Benchmarking deep multivariate time-series anomaly detection." In: *Transactions on Machine Learning Research* (2023).

[416] Benyou Wang, Lifeng Shang, Christina Lioma, Xin Jiang, Hao Yang, Qun Liu, and Jakob Grue Simonsen. "On position embeddings in bert." In: *International conference on learning representations*. 2020.

[417] Chengyu Wang, Kui Wu, Tongqing Zhou, Guang Yu, and Zhiping Cai. "Tsagen: synthetic time series generation for kpi anomaly detection." In: *IEEE Transactions on Network and Service Management* 19.1 (2021), pp. 130–145.

[418] Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. "Micn: Multi-scale local and global context modeling for long-term series forecasting." In: *The eleventh international conference on learning representations*. 2023.

[419] Qi Wang, Xiao Zhang, Mingyi Li, Yuan Yuan, Mengbai Xiao, Fuzhen Zhuang, and Dongxiao Yu. "TAMO: Fine-Grained Root Cause Analysis via Tool-Assisted LLM Agent with Multi-Modality Observation Data." In: *arXiv preprint arXiv:2504.20462* (2025).

[420] Qing Wang, Wubai Zhou, Chunqiu Zeng, Tao Li, Larisa Shwartz, and Genady Ya Grabarnik. "Constructing the knowledge base for cognitive it service management." In: *2017 IEEE International Conference on Services Computing (SCC)*. IEEE. 2017, pp. 410–417.

[421] Yu Wang, Yinxing Shen, Shiwen Mao, Xin Chen, and Hualei Zou. "LASSO and LSTM integrated temporal model for short-term solar intensity forecasting." In: *IEEE Internet of Things Journal* 6.2 (2018), pp. 2933–2944.

[422] Zhengwei Wang, Qi She, and Tomas E Ward. "Generative adversarial networks in computer vision: A survey and taxonomy." In: *ACM Computing Surveys (CSUR)* 54.2 (2021), pp. 1–38. URL: https://doi.org/10.1145/3439723.

[423] Ruofeng Wen, Kari Torkkola, Balakrishnan Narayanaswamy, and Dhruv Madeka. "A multi-horizon quantile recurrent forecaster." In: *arXiv preprint arXiv:1711.11053* (2017). URL: https://doi.org/10.48550/arXiv.1711.11053.

[424] Philipp Wenig, Stefan Schmidl, and Thilo Papenbrock. "TimeEval: A Benchmarking Toolkit for Time Series Anomaly Detection Algorithms." In: *Proceedings of the VLDB Endowment (PVLDB)* 15.13 (2022), pp. 3678–3681.

[425] Magnus Wiese, Robert Knobloch, Ralf Korn, and Peter Kretschmer. "Quant GANs: deep generation of financial time series." In: *Quantitative Finance* 20.9 (2020), pp. 1419–1440. URL: https://doi.org/10.1080/14697688.2020.1730426.

[426] Peter R Winters. "Forecasting sales by exponentially weighted moving averages." In: *Management science* 6.3 (1960), pp. 324–342.

[427] Mark Woike, Ali Abdul-Aziz, and Michelle Clem. "Structural health monitoring on turbine engines using microwave blade tip clearance sensors." In: *Smart Sensor Phenomena, Technology, Networks, and Systems Integration 2014*. Vol. 9062. SPIE. 2014, pp. 167–180.

[428] Lawrence Wong, Dongyu Liu, Laure Berti-Equille, Sarah Alnegheimish, and Kalyan Veeramachaneni. "AER: Auto-encoder with regression for time series anomaly detection." In: *2022 IEEE International Conference on Big Data (Big Data)*. IEEE. 2022, pp. 1152–1161.

[429] Fei Wu, Hongjian Wang, and Zhenhui Li. "Interpreting traffic dynamics using ubiquitous urban data." In: *Proceedings of the 24th ACM SIGSPATIAL international conference on advances in geographic information systems*. 2016, pp. 1–4. URL: https://doi.org/10.1145/2996913.2996962.

[430] Guangqiang Wu and Fu Zhang. "Multivariate Time Series Anomaly Detection using DiffGAN Model." In: *arXiv preprint arXiv:2501.01591* (2025).

[431] Haixu Wu, Tianzhou Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. "TimesNet: Temporal 2D-variation modeling for general time series analysis." In: *arXiv preprint arXiv:2210.02186* (2022).

[432] Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting." In: *Advances in neural information processing systems* 34 (2021), pp. 22419–22430. URL: https://doi.org/10.48550/arXiv.2106.13008.

[433] Renjie Wu and Eamonn J. Keogh. "Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress (Extended Abstract)." In: *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, May 2022, 1479–1480. DOI: 10.1109/icde53745.2022.00116. URL: http://dx.doi.org/10.1109/icde53745.2022.00116.

[434] Wentai Wu, Ligang He, Weiwei Lin, Yi Su, Yuhua Cui, Carsten Maple, and Stephen Jarvis. "Developing an unsupervised real-time anomaly detection scheme for time series with multi-seasonality." In: *IEEE Transactions on Knowledge and Data Engineering* 34.9 (2022), pp. 4147–4160.

[435] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. "Unsupervised Feature Learning via Non-parametric Instance Discrimination." In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 2018, 3733–3742. DOI: 10.1109/cvpr.2018.00393. URL: http://dx.doi.org/10.1109/cvpr.2018.00393.

[436] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. "Connecting the dots: Multivariate time series forecasting with graph neural networks." In: *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 2020, pp. 753–763. URL: https://doi.org/10.1145/3394486.3403118.

[437] Jiang Xiao, Zhuang Xiong, Song Wu, Yusheng Yi, Hai Jin, and Kan Hu. "Disk Failure Prediction in Data Centers via Online Learning." In: *Proceedings of the 47th International Conference on Parallel Processing*. ICPP 2018. ACM, Aug. 2018, pp. 1–10. DOI: 10.1145/3225058.3225106. URL: http://dx.doi.org/10.1145/3225058.3225106.

[438] Shiwang Xing, Jianwei Niu, and Tao Ren. "GCFormer: Granger Causality based Attention Mechanism for Multivariate Time Series Anomaly Detection." In: *2023 IEEE International Conference on Data Mining (ICDM)*. IEEE, Dec. 2023, 1433–1438. DOI: 10.1109/icdm58522.2023.00187. URL: http://dx.doi.org/10.1109/icdm58522.2023.00187.

[439] Chang Xu, Gang Wang, Xiaoguang Liu, Dongdong Guo, and Tie-Yan Liu. "Health status assessment and failure prediction for hard drives with recurrent neural networks." In: *IEEE Transactions on Computers* 65.11 (2016), pp. 3502–3508.

[440] Congyuan Xu, Jizhong Shen, and Xin Du. "Low-rate DoS attack detection method based on hybrid deep neural networks." In: *Journal of Information Security and Applications* 60 (2021), p. 102879.

[441] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. "Self-attention with functional time representation learning." In: *Advances in neural information processing systems* 32 (2019).

[442] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. "Inductive representation learning on temporal graphs." In: *arXiv preprint arXiv:2002.07962* (2020).

[443] Hao Xu, Guansong Pang, Yibing Wang, and Yu Wang. "Deep Isolation Forest for Anomaly Detection." In: *IEEE Transactions on Knowledge and Data Engineering* 35.12 (2023), pp. 12591–12604.

[444] Hao Xu, Yu Wang, Shuai Jian, Qiang Liao, Yibing Wang, and Guansong Pang. "Calibrated one-class classification for unsupervised time series anomaly detection." In: *IEEE Transactions on Knowledge and Data Engineering* (2024).

[445] Haowen Xu et al. "Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications." In: *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*. WWW '18. ACM Press, 2018, pp. 187–196. DOI: 10.1145/3178876.3185996. URL: http://dx.doi.org/10.1145/3178876.3185996.

[446] Jingkai Xu, Haixu Wu, Jianmin Wang, and Mingsheng Long. "Anomaly Transformer: Time series anomaly detection with association discrepancy." In: *arXiv preprint arXiv:2110.02642* (2021).

[447] Jun Xu, Rouhollah Rahmatizadeh, Ladislau Bölöni, and Damla Turgut. "Real-time prediction of taxi demand using recurrent neural networks." In: *IEEE Transactions on Intelligent Transportation Systems* 19.8 (2017), pp. 2572–2581. URL: https://doi.org/10.1109/TITS.2017.2755684.

[448] Xin Xu, Hongbo Zhao, Haoqiang Liu, and Hua Sun. "LSTM-Gan-xgboost based anomaly detection algorithm for time series data." In: *2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan)*. IEEE. 2020, pp. 334–339.

[449] Ying Xu and Dongsheng Li. "Incorporating graph attention and recurrent architectures for city-wide taxi demand prediction." In: *ISPRS International Journal of Geo-Information* 8.9 (2019), p. 414. URL: https://doi.org/10.3390/ijgi8090414.

[450] Jianbo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiaoli Li, Shonali Krishnaswamy, et al. "Deep convolutional neural networks on multichannel time series for human activity recogni-

tion." In: *Ijcai*. Vol. 15. Buenos Aires, Argentina. 2015, pp. 3995–4001.

[451] Jingyu Yang, Zuogong Yue, and Ye Yuan. "Deep probabilistic graphical modeling for robust multivariate time series anomaly detection with missing data." In: *Reliability Engineering & System Safety* 238 (Oct. 2023), p. 109410. ISSN: 0951-8320. DOI: 10.1016/j.ress.2023.109410. URL: http://dx.doi.org/10.1016/j.ress.2023.109410.

[452] Yiyuan Yang, Chaoli Zhang, Tian Zhou, Qingsong Wen, and Liang Sun. "DCdetector: Dual Attention Contrastive Representation Learning for Time Series Anomaly Detection." In: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD '23. ACM, Aug. 2023, 3033–3045. DOI: 10.1145/3580305.3599295. URL: http://dx.doi.org/10.1145/3580305.3599295.

[453] Zheng Yang, Linyue Liu, Ning Li, and Junwei Tian. "Time Series Forecasting of Motor Bearing Vibration Based on Informer." In: *Sensors (Basel, Switzerland)* 22 (2022). DOI: 10.3390/s22155858.

[454] Huaxiu Yao, Xianfeng Tang, Hua Wei, Guanjie Zheng, and Zhenhui Li. "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction." In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 01. 2019, pp. 5668–5675. URL: https://doi.org/10.1609/aaai.v33i01.33015668.

[455] Huaxiu Yao, Fei Wu, Jintao Ke, Xianfeng Tang, Yitian Jia, Siyu Lu, Pinghua Gong, Jieping Ye, and Zhenhui Li. "Deep multi-view spatial-temporal network for taxi demand prediction." In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018, pp. 2588–2595. URL: https://doi.org/10.48550/arXiv.1802.08714.

[456] Yueyue Yao, Jianghong Ma, Shanshan Feng, and Yunming Ye. "SVD-AE: An asymmetric autoencoder with SVD regularization for multivariate time series anomaly detection." In: *Neural Networks* 170 (Feb. 2024), pp. 535–547. ISSN: 0893-

6080. DOI: 10.1016/j.neunet.2023.11.023. URL: http://dx.doi.org/10.1016/j.neunet.2023.11.023.

[457] Véronne Yepmo, Grégory Smits, and Olivier Pivert. "Anomaly explanation: A review." In: *Data & Knowledge Engineering* 137 (2022), p. 101946.

[458] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. "Time-series generative adversarial networks." In: *Advances in neural information processing systems* 32 (2019). URL: https://proceedings.neurips.cc/paper_files/paper/2019/file/c9efe5f26cd17ba6216bbe2a7d26d490-Paper.pdf.

[459] Sungmin You, Baek Hwan Cho, Young-Min Shon, Dae-Won Seo, and In Young Kim. "Semi-supervised automatic seizure detection using personalized anomaly detecting variational autoencoder with behind-the-ear EEG." In: *Computer Methods and Programs in Biomedicine* 213 (2022), p. 106542.

[460] W. J. Youden. "Index for rating diagnostic tests." In: *Cancer* 3.1 (1950), pp. 32–35. ISSN: 1097-0142. DOI: 10.1002/1097-0142(1950)3:1<32::aid-cncr2820030106>3.0.co;2-3. URL: http://dx.doi.org/10.1002/1097-0142(1950)3:1<32::aid-cncr2820030106>3.0.co;2-3.

[461] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. "Recent trends in deep learning based natural language processing." In: *ieee Computational intelligenCe magazine* 13.3 (2018), pp. 55–75.

[462] Bing Yu, Haoteng Yin, and Zhanxing Zhu. "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting." In: *arXiv preprint arXiv:1709.04875* (2017). URL: https://doi.org/10.24963/ijcai.2018/505.

[463] Chuanjin Yu, Yongle Li, Yulong Bao, Haojun Tang, and Guanghao Zhai. "A novel framework for wind speed prediction based on recurrent neural networks and support vector machine." In: *Energy Conversion and Management* 178 (2018), pp. 137–145.

[464] Fisher Yu and Vladlen Koltun. "Multi-scale context aggregation by dilated convolutions." In: *arXiv preprint arXiv:1511.07122* (2015).

[465] Jing Yuan, Yu Zheng, Liuhang Zhang, XIng Xie, and Guangzhong Sun. "Where to find my next passenger." In: *Proceedings of the 13th international conference on Ubiquitous computing*. 2011, pp. 109–118. URL: http://dx.doi.org/10.1145/2030112.2030128.

[466] Zahra Zamanzadeh Darban, Geoffrey I Webb, Shirui Pan, Charu Aggarwal, and Mahsa Salehi. "Deep learning for time series anomaly detection: A survey." In: *ACM Computing Surveys* (2022).

[467] Luca Zancato, Alessandro Achille, Giovanni Paolini, Alessandro Chiuso, and Stefano Soatto. "STRIC: Stacked Residuals of Interpretable Components for Time Series Anomaly Detection." In: (2021).

[468] Fanyu Zeng, Mengdong Chen, Cheng Qian, Yanyang Wang, Yijun Zhou, and Wenzhong Tang. "Multivariate time series anomaly detection with adversarial transformer architecture in the Internet of Things." In: *Future Generation Computer Systems* 144 (July 2023), 244–255. ISSN: 0167-739X. DOI: 10.1016/j.future.2023.02.015. URL: http://dx.doi.org/10.1016/j.future.2023.02.015.

[469] Jun Zhan, Siqi Wang, Xiandong Ma, Chengkun Wu, Canqun Yang, Detian Zeng, and Shilin Wang. "Stgat-Mad: Spatial-Temporal Graph Attention Network For Multivariate Time Series Anomaly Detection." In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, May 2022, pp. 3568–3572. DOI: 10.1109/icassp43922.2022.9747274. URL: http://dx.doi.org/10.1109/icassp43922.2022.9747274.

[470] Biao Zhang, Chao Song, Xu chu Jiang, and Ying Li. "Electricity price forecast based on the STL-TCN-NBEATS model." In: *Heliyon* 9 (2023). DOI: 10.1016/j.heliyon.2023.e13029.

[471] Chunkai Zhang, Wei Zuo, Shaocong Li, Xuan Wang, Peiyi Han, and Chuanyi Liu. "Reconstruct Anomaly to Normal: Adversarially Learned and Latent Vector-Constrained Autoencoder for Time-Series Anomaly Detection." In: *PRICAI 2021: Trends in Artificial Intelligence*. Springer International Publishing, 2021, 515–529. ISBN: 9783030893637. DOI: 10.1007/978-3-030-89363-7_39. URL: http://dx.doi.org/10.1007/978-3-030-89363-7_39.

[472] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, and Nitesh V Chawla. "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.

[473] Guangyao Zhang, Xin Gao, Lei Wang, Bing Xue, Shiyuan Fu, Jiahao Yu, Zijian Huang, and Xu Huang. "Probabilistic autoencoder with multi-scale feature extraction for multivariate time series anomaly detection." In: *Applied Intelligence* 53.12 (Nov. 2022), pp. 15855–15872. ISSN: 1573-7497. DOI: 10.1007/s10489-022-04324-3. URL: http://dx.doi.org/10.1007/s10489-022-04324-3.

[474] Junbo Zhang, Yu Zheng, and Dekang Qi. "Deep spatio-temporal residual networks for citywide crowd flows prediction." In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 31. 2017, pp. 1655–1661. URL: https://doi.org/10.48550/arXiv.1610.00081.

[475] Lingzhe Zhang, Tong Jia, Mengxi Jia, and Ying Li. "LogDB: Multivariate Log-based Failure Diagnosis for Distributed Databases (Extended from MultiLog)." In: *arXiv preprint arXiv:2505.01676* (2025).

[476] Lingzhe Zhang, Tong Jia, Mengxi Jia, Yong Yang, Zhonghai Wu, and Ying Li. "A Survey of AIOps for Failure Management in the Era of Large Language Models." In: *arXiv preprint arXiv:2406.11213* (2024).

[477] Shenglin Zhang, Sibo Xia, Wenzhao Fan, Binpeng Shi, Xiao Xiong, Zhenyu Zhong, Minghua Ma, Yongqian Sun, and Dan Pei. "Failure Diagnosis in Microservice Systems: A Comprehensive Survey and Analysis." In: *ACM Transactions on Software Engineering and Methodology* (Jan. 2025). ISSN: 1557-7392. DOI: 10.1145/3715005. URL: http://dx.doi.org/10.1145/3715005.

[478] W. Zhang, Z. Wu, Xiaojun Zeng, and Changhui Zhu. "An ensemble dynamic self-learning model for multiscale carbon price forecasting." In: *Energy* (2022). DOI: 10.1016/j.energy.2022.125820.

[479] Weiqi Zhang, Chen Zhang, and Fugee Tsung. "GRELEN: Multivariate Time Series Anomaly Detection from the Perspective of Graph Relational Learning." In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*. IJCAI-2022. International Joint Conferences on Artificial Intelligence Organization, July 2022, 2390–2397. DOI: 10.24963/ijcai.2022/332. URL: http://dx.doi.org/10.24963/ijcai.2022/332.

[480] Yuxin Zhang, Yiqiang Chen, Jindong Wang, and Zhiwen Pan. "Unsupervised deep anomaly detection for multi-sensor time-series signals." In: *IEEE Transactions on Knowledge and Data Engineering* 35.2 (2021), pp. 2118–2132.

[481] Ying Zhao, Xiang Liu, Siqing Gan, and Weimin Zheng. "Predicting disk failures with HMM-and HSMM-based approaches." In: *Advances in Data Mining. Applications and Theoretical Aspects: 10th Industrial Conference, ICDM 2010, Berlin, Germany, July 12-14, 2010. Proceedings 10*. Springer. 2010, pp. 390–404.

[482] Shuai Zheng, Kosta Ristovski, Ahmed Farahat, and Chetan Gupta. "Long Short-Term Memory Network for Remaining Useful Life estimation." In: *2017 IEEE International Conference on Prognostics and Health Management (ICPHM)*. IEEE, June 2017, 88–95. DOI: 10.1109/icphm.2017.7998311. URL: http://dx.doi.org/10.1109/icphm.2017.7998311.

[483] Jie Zhong, Enguang Zuo, Chen Chen, Cheng Chen, Junyi Yan, Tianle Li, and Xiaoyi Lv. "A Masked Attention Network with Query Sparsity Measurement for Time Series Anomaly Detection." In: *2023 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, July 2023, 2741–2746. DOI: 10.1109/icme55011.2023.00466. URL: http://dx.doi.org/10.1109/icme55011.2023.00466.

[484] Zhenyu Zhong, Qiliang Fan, Jiacheng Zhang, Minghua Ma, Shenglin Zhang, Yongqian Sun, Qingwei Lin, Yuzhi Zhang, and Dan Pei. "A Survey of Time Series Anomaly Detection Methods in the AIOps Domain." In: *arXiv preprint arXiv:2308.00393* (2023).

[485] Bin Zhou, Shenghua Liu, Bryan Hooi, Xueqi Cheng, and Jing Ye. "Beatgan: Anomalous rhythm detection using adversarially generated time series." In: *IJCAI*. Vol. 2019. 2019, pp. 4433–4439.

[486] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. "Informer: Beyond efficient transformer for long sequence time-series forecasting." In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 35. 12. 2021, pp. 11106–11115. URL: https://doi.org/10.48550/arXiv.2012.07436.

[487] Qihang Zhou, Jiming Chen, Haoyu Liu, Shibo He, and Wenchao Meng. "Detecting Multivariate Time Series Anomalies with Zero Known Label." In: *Proceedings of the AAAI Conference on Artificial Intelligence* 37.4 (June 2023), 4963–4971. ISSN: 2159-5399. DOI: 10.1609/aaai.v37i4.25623. URL: http://dx.doi.org/10.1609/aaai.v37i4.25623.

[488] Justus Zipfel, Felix Verworner, Marco Fischer, Uwe Wieland, Mathias Kraus, and Patrick Zschech. "Anomaly detection for industrial quality assurance: A comparative evaluation of unsupervised deep learning models." In: *Computers & Industrial Engineering* 177 (2023), p. 109045.

[489] Banca d'Italia. *Payment Systems - Framework and definitions*. URL: https://www.bancaditalia.it/compiti/sispaga-mercati/sistemi-pagamenti/index.html?com.dotmarketing.htmlpage.language=1 (visited on 10/09/2024).

[490] Banca d'Italia. *Denmark joins T2 and TIPS to fully integrate Danish krone in Eurosystem's payment services*. 2024. URL: https://www.bancaditalia.it/media/notizia/denmark-joins-t2-and-tips-to-fully-integrate-danish-krone-in-eurosystem-s-payment-services/ (visited on 03/21/2024).

[491] *ksqlDB*. https://github.com/confluentinc/ksql.