Time Series Foundation Models for Operational Support in Geothermal Systems

Bridging the Gap between Advanced AI and Energy Infrastructure

Master Thesis

Zeryab Alam

Delft University of Technology

TU Delft

Time Series Foundation Models for Operational Support in Geothermal Systems

Bridging the Gap between Advanced AI and Energy Infrastructure

by

Zeryab Alam

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday June 16, 2026 at 01:00 PM.

Student number: 5486548
Project duration: November 3, 2025 – June 16, 2026
Thesis committee:

Dr. A. Anand,	TU Delft, Chair
Dr. K. Atasu,	TU Delft, Core Member
Dr. J. Decouchant,	TU Delft, Core Member
Dr. P. Shoeibi Omrani,	TNO, External Member

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Preface

This thesis marks the conclusion of my Master of Science in Computer Science at Delft University of Technology, a journey that began in November 2025 and ended in June 2026. Looking back, this milestone represents the final chapter of my five-year period at TU Delft, which started with my Bachelor's degree in 2021. That initial phase was very demanding and highly stressful, but also a very rewarding rollercoaster. Transitioning into the Master's programme in 2024 allowed me to broaden my horizons, exploring everything from scalable distributed systems to specialized machine learning, ultimately leading to the research presented in this thesis.

First and foremost, I must express my deepest gratitude to God. He has always given me the strength to persevere through the most challenging periods of my life and has always shone light upon my path. This achievement would quite simply not be possible without Him.

I want to extend my heartfelt thanks to my company supervisor, Dr. Pejman Shoeibi Omrani at TNO. Thank you for believing in me during that summer 2025 interview and giving me the opportunity to dive into advanced industrial AI applications. Coming from a pure Computer Science background, I had little to no clue how geothermal systems worked. I am incredibly grateful for your patience in explaining the mechanics, the industry bottlenecks, and the entire domain to me, all while gracefully handling my endless stream of daily questions.

I am equally indebted to my university thesis advisor, Dr. Kubilay Atasu. My interest in this domain was sparked during your Scalable Learning research course, and I am glad I persuaded you to supervise my thesis. Thank you for your invaluable insights, especially when the initial results looked bleak. Your guidance on how to interpret and frame setbacks completely shaped the direction of this research, and your unwavering support was instrumental as I worked on this thesis.

This achievement belongs to my family and friends as much as it does to me. To my family, thank you for financing my studies from start to finish, and for providing the emotional fallback that kept me going. Mom, thank you for your constant, protective care and for always ensuring I made it home safe. Dad, your insights and guidance shaped every major decision I faced. My brothers, thank you for always being there and listening to me complain about things completely irrelevant to you. And to my friends, thank you for the solidarity, both in the late nights we spent cramming for exams and the late nights we spent playing games to escape the stress. You made this entire journey meaningful.

Zeryab Alam
Delft, June 2026

Contents

1

Introduction

Climate change has become one of the most significant challenges of the 21st century [39]. It is primarily driven by greenhouse gas emissions resulting from the combustion of fossil fuels for energy production in homes, industry, and transport [40]. The Industrial Revolution of the 19th century, while a major milestone in human progress, also established a long-term dependence on carbon-intensive energy systems that now contribute significantly to global warming [17]. At the same time, global energy demand continues to rise [18] due to population growth, industrial development, making the transition to sustainable energy sources increasingly urgent. Renewable energy sources, particularly solar and wind energy, have therefore seen rapid adoption [5] as alternatives to fossil fuels. However, their effectiveness as a complete replacement remains uncertain, and their inherent intermittency — due to dependence on weather conditions such as sunlight and wind — introduces variability that poses challenges for maintaining a stable and reliable power grid [3].

Geothermal energy [20] is uniquely positioned to address some of the key limitations of solar and wind power due to its ability to provide continuous and weather-independent energy. It utilizes the Earth's naturally occurring subsurface heat, which can be converted into electrical power and thermal energy for various applications [12]. As a renewable energy source when managed sustainably, geothermal energy has been shown to have significant potential, with some estimates suggesting it could contribute at the terawatt scale [11] under favorable deployment scenarios. However, its utilization also introduces significant engineering challenges. Geothermal energy production relies on complex infrastructure, including extraction systems, heat exchangers, power conversion units, and re-injection processes. To maintain reliable output, these plants must operate continuously, which places constant stress on their components. Over time, this operational stress leads to gradual degradation, reducing system efficiency and performance [34] [32]. If not detected early, such degradation can result in severe failures and significant economic losses [43], particularly given that many geothermal plants currently operate within narrow profitability margins [14]. Consequently, continuous monitoring and maintenance of geothermal systems is essential to ensure safe and efficient long-term operation.

The industry has historically relied on simple threshold-based monitoring systems for condition monitoring and fault detection [29]. While these approaches are easy to interpret and implement, they are limited in their ability to detect subtle changes in system behavior and provide little to no predictive capability. More recently, some operators have adopted traditional machine learning models to learn system behavior directly from data [54]. However, these approaches [24] typically require large amounts of labeled training data [52], are highly task-specific, and often struggle to generalize across different plant sites [22]. These limitations are particularly critical in geothermal energy systems, where failure events are rare and labeled examples of faults are scarce. In addition, concerns around data privacy and operational security often prevent operators from sharing plant-level sensor data, resulting in fragmented and siloed datasets [53]. Together, these constraints in data availability, generalization, and privacy demand a fundamental departure from conventional machine learning workflows.

Foundation models [6] represent precisely this kind of departure, offering a new paradigm built around

generalization rather than task-specific training. Rather than learning from scratch on a narrow dataset, these models are pretrained on large, diverse datasets, allowing them to develop broad representations that transfer across tasks with minimal additional training. This shift was largely enabled by transformer architectures [47], which excel at capturing long-range dependencies in sequential data. The practical consequence is significant: downstream adaptation no longer requires extensive labeled datasets or task-specific training pipelines — a particular advantage in industrial settings like geothermal energy operations, where data is scarce, fragmented, and rarely shared across sites. That said, the success foundation models have demonstrated in language and vision does not automatically transfer to time series data due to fundamental differences in data structure, including continuous-valued inputs, strong temporal dependencies, and non-stationary dynamics [55] [30].

Adapting foundation models to time series data is non-trivial. Unlike text or images, time series lack a natural tokenization process, and the diversity of sampling rates, scales, and temporal dynamics across domains makes it difficult for any single model to generalize broadly. Time Series Foundation Models (TSFMs) [26] have emerged as a direct response to these challenges, developing architectural strategies that aim to capture general temporal representations applicable across this diversity. Despite their growing momentum, however, TSFMs remain largely unexplored in complex industrial environments. This gap is partly structural: most early TSFMs were designed exclusively for univariate settings, making them ill-suited for industrial condition monitoring, where systems are inherently multivariate [4][45]. Recent architectural advances have begun to change this [50], yet their suitability for geothermal condition monitoring remains largely unexplored. It is unclear how well they handle the unique operational dynamics of such systems, how they compare against established forecasting and anomaly detection baselines, and to what extent their zero-shot capabilities hold under the specific constraints and characteristics of geothermal data.

This thesis investigates these questions through a systematic evaluation of Time Series Foundation Models for geothermal condition monitoring, spanning forecasting and anomaly detection tasks. All models are evaluated in a zero-shot setting, without any task-specific training or fine-tuning, to assess how well pretrained representations transfer to a specialized industrial domain.

The overarching research question addressed in this thesis is:

To what extent can Time Series Foundation Models serve as a practical and effective zero-shot solution for condition monitoring in geothermal energy systems?

The primary contribution of this thesis is the first systematic evaluation of Time Series Foundation Models in the geothermal energy domain. Through zero-shot forecasting and anomaly detection experiments, we derive insights into how well TSFMs transfer to a complex industrial setting, what architectural properties drive their generalization, and where their current limitations lie, findings that carry broader implications for the application of foundation models in industrial time series environments.

The thesis is structured as follows. Chapter 2 presents the theoretical background and literature survey, introducing time series data, machine and deep learning approaches, and foundation models for time series. Chapter 3 presents the main scientific research article included in this thesis, which evaluates Time Series Foundation Models for forecasting and anomaly detection in geothermal operations under zero-shot settings. Finally, Chapter 4 concludes the thesis and outlines directions for future research.

2

Background

2.1. Time Series Analysis

2.1.1. Time Series Data

A time series is an ordered sequence of observations indexed by time. Formally, it is represented as $\{x_t\}_{t=1}^{T}$, where $x_t \in \mathbb{R}$ denotes the observation recorded at time step t. Time series data is often *temporally dependent*: the value observed at time t is causally influenced by preceding observations, such that x_t is not independent of $x_{t-1}, x_{t-2}, ...$ This distinguishes time series from independently and identically distributed (i.i.d.) data and motivates the use of models that explicitly capture sequential structure.

2.1.2. Univariate and Multivariate Time Series

A univariate time series tracks a single variable over time, while a multivariate time series consists of multiple variables recorded simultaneously, $x_t = (x_t^{(1)}, x_t^{(2)}, ..., x_t^{(N)}) \in \mathbb{R}^N$. In practice, the variables within a multivariate series are often interdependent: the evolution of one variable may or may not influence others. In industrial systems, for instance, pump frequency directly influences flow rate, which in turn affects downstream temperatures and pressures. Capturing these cross-variable dependencies is essential for accurate modeling.

2.1.3. Properties of Time Series Data

Time series data typically consists of four main properties [25], as illustrated in Figure 2.1. **Trend** refers to the long term directional movement of the series: a sustained increase or decrease over time. **Seasonality** refers to periodic, repeating patterns that occur regularly such as increased energy consumption during winter or higher sales during holidays. **Cyclicity** refers to the long term fluctuations that occur around the underlying trend but do not follow a fixed periodic schedule. **Irregularity** refers to the irregular fluctuations that cannot be represented by trend, seasonality or cyclicity.

A time series is **stationary** if its mean and variance remain constant over time. Traditional forecasting methods typically assume stationarity. However, this is rarely observed in real-world industrial data, usually because it is heavily influenced by external factors like operator controls.

2.1.4. Covariates

Time series data is often influenced by external variables, known as **covariates** or exogenous variables. These often include features such as holidays for sales context or operator controlled inputs such as pump frequency in the geothermal domain. Covariates can be past known values or known future values. Incorporating these covariates when forecasting can result in a much more stronger performance compared to relying on just the target's temporal structure instead.

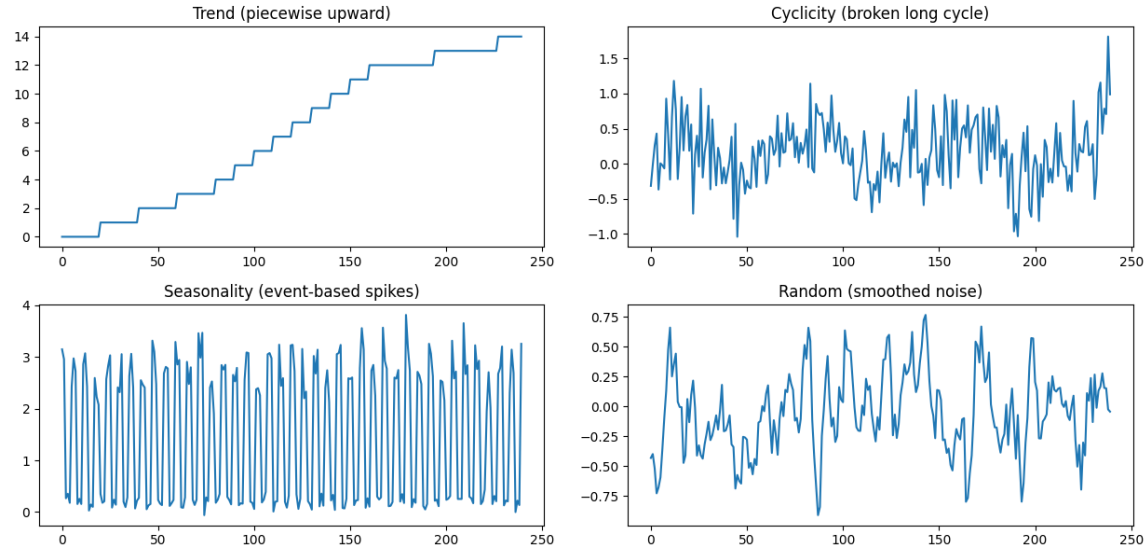


Figure 2.1: Visual representation of the four properties of time series data: Trend, Seasonality, Cyclicity, and Irregularity.

2.1.5. Forecasting

Forecasting involves estimating future values of a time series based on past observations. Given a context window of length L, a forecasting model produces predictions over a future horizon of length H:

$$\hat{x}_{t+1}, \hat{x}_{t+2}, ..., \hat{x}_{t+H} = f(x_{t-L+1}, x_{t-L+2}, ..., x_t). \tag{2.1}$$

The choice of horizon H determines how far ahead predictions are made, although, longer horizons are typically more challenging as uncertainty accumulates.

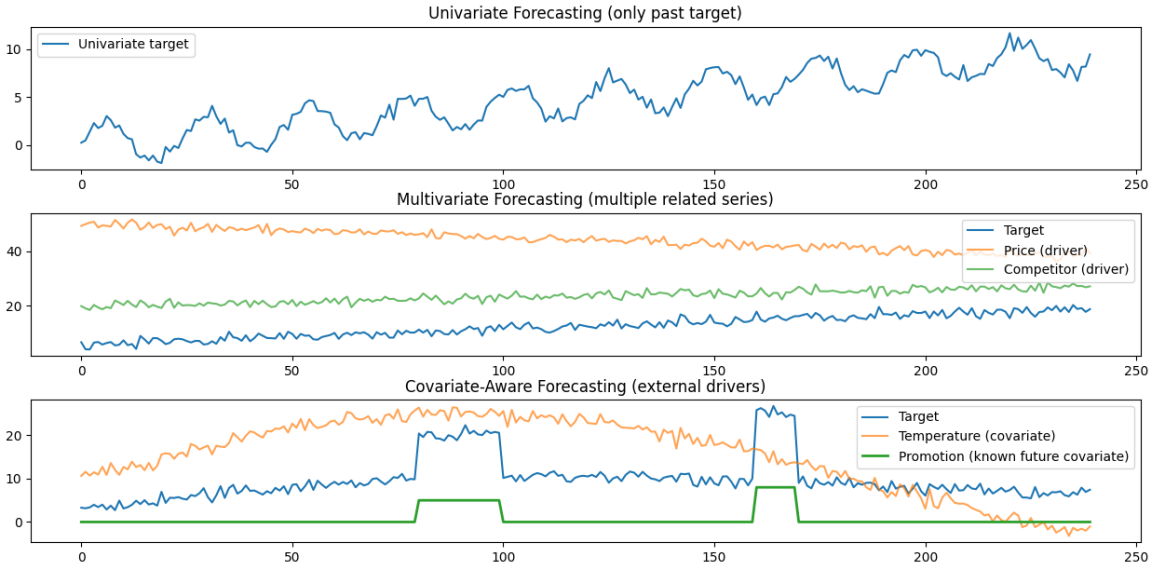


Figure 2.2: Visual representation of the three types of forecasting: Univariate, Multivariate, and Covariate-aware

The task differs based on the information available to the model, as illustrated in Figure 2.2. **Univariate forecasting** considers only the historical values of the target variable to predict its future trajectory. **Multivariate forecasting** models multiple time series jointly, enabling the learning of relationships across related variables. **Covariate-aware forecasting** [35] predicts a target variable using both its historical values and additional covariates, which may include past as well as future-known observations that influence the target.

2.2. Statistical Models for Time Series

2.2.1. Auto-Regressive (AR) Models

AR models forecast future values using a linear combination of their own past values, operating on the assumption that the future depends directly on the history [8]. This dependency is controlled by defining how many past values the model can use to make its predictions.

AR models assume that the underlying time series is stationary and linear relationships exist between past and future values. Due to this, they are generally well-suited for simple temporal dynamics but struggle with complex non-linear patterns.

2.2.2. Moving Average (MA) Models

Time series data may consist of noise in individual points, which disrupts AR models because a noisy observation at time t becomes an input for predictions at all subsequent steps $t+1, t+2, ..., t+H$, causing error propagation. To counter this short-term noise, MA models smooth time series data using a sliding window. By replacing values with the average of past observations, this approach acts as a filter to stabilize underlying trends [33].

Similar to AR models, MA models also rely on the assumptions of stationarity and linearity, and thus face the same limitations of being unable to model complex long-term temporal patterns.

2.2.3. ARMA and ARIMA

The Auto-Regressive Moving Average (ARMA) model [56] combines both AR and MA components, allowing it to capture historical trends while mitigating sudden shocks. This makes it more expressive that either AR or MA alone, while still maintaining a simple linear structure. However, ARMA still requires the time series to be stationary.

The Auto-Regressive Integrated Moving Average (ARIMA) model [44] extends ARMA by introducing an integration step, where differencing is applied to the data to remove trends and enforce stationarity. While this improves flexbility for real-world data, ARIMA remains fundamentally linear and assumes that relationships in the data can be captured through differencing and past dependencies. It does not incorporate external variables, further limiting its performance in environments where external variables play an important role.

2.2.4. SARIMA and SARIMAX

Seasonal ARIMA (SARIMA) [46] extends ARIMA by explicitly modeling repeating seasonal patterns through seasonal auto-regressive and seasonal differencing components, allowing the model to capture periodic structures such as daily, weekly, or yearly cycles in the data.

SARIMAX [1] further extends this framework by incorporating external variables through a linear regression component. These external variables, as discussed earlier, play a significant role in the dynamics of the systems and can heavily influence the target series.

These models, however, inherit the same limitations concerning ARIMA. They assume existing seasonal structure, and may struggle with complex non-linear temporal structures. Additionally, their performance typically degrades in high-dimensional settings where many external variables are present.

2.3. Deep Learning for Time Series

2.3.1. Recurrent Neural Networks

Since time series data is sequential, Recurrent Neural Networks (RNNs) [41] provide a strong approach for learning patterns within the data. Traditional feed-forward neural networks assume that each datapoint is independent. However, this assumption does not hold for time series data, where observations are heavily influenced by preceding datapoints. RNNs are specifically designed to model these temporal dependencies by maintaining a hidden state that acts as memory and evolves over time.

At each time step t, the model take an input vector x_t and update its hidden state h_t as follows:

$$h_t = \tanh(W_x x_t + W_h h_{t-1} + b_h) \tag{2.2}$$

where W_x is the input-to-hidden weight matrix, W_h is the hidden-to-hidden weight matrix, and b_h is a bias term. The hidden state h_t acts as a compact memory representation of all previously observed inputs up to time t.

The output at each time step is computed as:

$$y_t = W_y h_t + b_y \tag{2.3}$$

where W_y is the hidden-to-output weight matrix and b_y is the output bias term. This formulation is visualized in Figure 2.3 over multiple timesteps and allows the model to generate a prediction at each time step while still preserving temporal context through the hidden state.

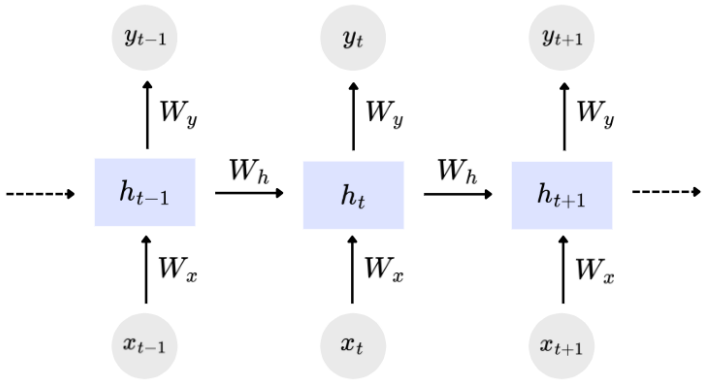


Figure 2.3: Sequential flow of a RNN across consecutive timesteps. The hidden state h_t continuously updates using the current input x_t and the previous hidden state h_{t-1} to map dependencies before generating the output y_t.

Standard RNNs, however, suffer from several limitations. Since information is propagated through a single hidden state across many timesteps, older information is usually overwritten. RNNs also often suffer from vanishing or exploding gradients, making it difficult to learn long-term dependencies. This results in important historical context being lost, limiting the model's effectiveness [13].

2.3.2. Long Short-Term Memory Networks

Long Short-Term Memory (LSTM) networks [15] were introduced to address the limitations of standard RNNs. Rather than compressing all historical context into a single, continuously overwritten hidden state, LSTMs introduce a dedicated cell state that preserves relevant information across long sequences.

The cell state controls the flow of information using three gating mechanisms: the input gate which controls what new information should be stored in the cell state, the forget gate which controls what information is retained from the previous cell state, and the output gate which controls what information should be sent to the hidden state. The overall architecture of a LSTM cell is illustrated in Figure 2.4.

Formally, at each time step t, state updates use the current input vector x_t, the previous hidden state h_{t-1}, and the previous cell state c_{t-1} to perform gating computations:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{2.4}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{2.5}$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{2.6}$$

where W and U represent the input-to-hidden and hidden-to-hidden weight matrices, b denotes the bias vectors and $\sigma(\cdot) = \frac{1}{1+e^{-(\cdot)}}$ is the sigmoid function ensuring all gate values lie in $(0, 1)$. The final cell state c_t is updated via a linear combination of the previous cell state and the new cell state:

$$c_t = (f_t \odot c_{t-1}) + (i_t \odot \tilde{c}_t) \tag{2.7}$$

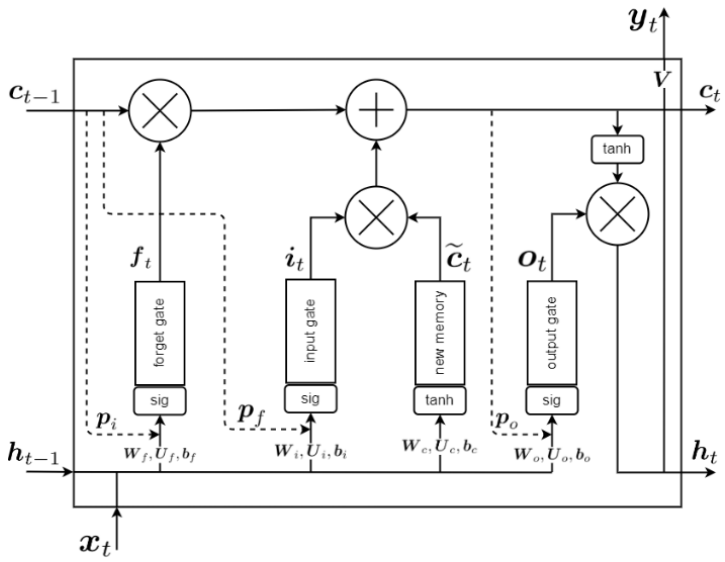


Figure 2.4: Internal gating mechanism of a single LSTM cell, adapted from Ghojogh and Ghodsi [13]. Element-wise multiplication ($\otimes$) and addition ($\oplus$) operators regulate how the forget gate f_t, input gate i_t, and output gate o_t modify the long-term memory cell state c_t and compute the recurrent hidden state h_t.

where $\odot$ denotes the Hadamard (element-wise) product. After the final cell state has been computed, the output gate computes information relevant to update the hidden state:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{2.8}$$

$$h_t = o_t \odot \tanh(c_t) \tag{2.9}$$

Finally, the output is computed via the hidden state:

$$y_t = V h_t + b_y \tag{2.10}$$

where V is the hidden-to-output weight matrix and y_t is the final output.

LSTMs are significantly more effective at capturing long temporal patterns compared to standard RNNs, and have been widely adopted for forecasting tasks. However, they remain inherently sequential, limiting parallelization during training and inference. Furthermore, LSTMs often require large amounts of training data to achieve strong performance and may still struggle with very long sequences [13]. These limitations motivated the development of attention-based architectures such as the Transformer.

2.3.3. Transformers and Self-Attention

To mitigate the limitations of RNNs, the Transformer architecture [47] was introduced, relying on the attention mechanism instead of recurrence to model relationships across sequences.

Attention allows the model to focus on the most relevant parts of the input sequence. Rather than compressing historical information into a single hidden state, attention allows the model to consider all prior steps and determine which ones are the most important for prediction.

Transformers implement this mechanism through self-attention, in which each step in a sequence attends to all others, as illustrated in Figure 2.5. Formally, given an input sequence represented as $X \in \mathbb{R}^{n \times d}$, where n is the sequence length and d is the hidden dimension, the model computes the query, key and value matrices (Q, K, and V) as

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V, \tag{2.11}$$

where $W_Q, W_K \in \mathbb{R}^{d \times d_q}$ and $W_V \in \mathbb{R}^{d \times d_v}$ are learnable projection matrices, and d_q denotes the dimensionality of the query and key vectors.

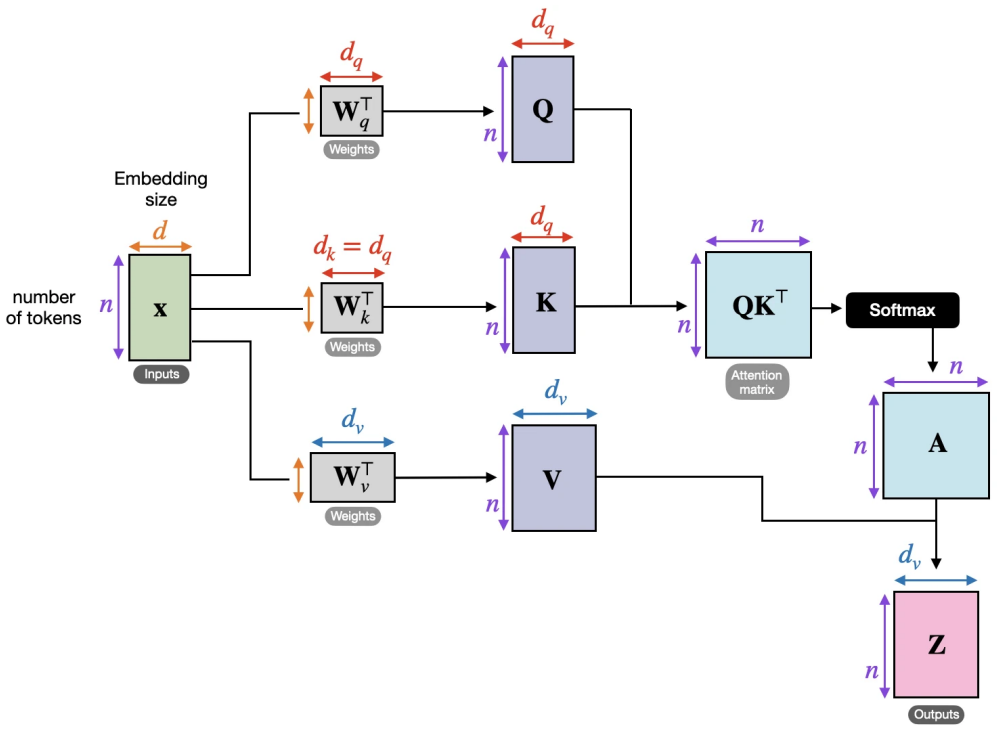


Figure 2.5: Visualization of the self-attention mechanism, adapted from Raschka [38].

The scaled attention weights, denoted as the matrix $A \in \mathbb{R}^{n \times n}$, are then calculated using a softmax activation:

$$A = \text{softmax}\left(\frac{QK^T}{\sqrt{d_q}}\right), \tag{2.12}$$

where QK^T computes dot-product similarity scores between queries and keys across timesteps, scaled by $1/\sqrt{d_q}$ to prevent vanishing gradients during training. The finalized attention matrix A is subsequently multiplied by the value matrix V to produce the final contextualized output matrix $Z \in \mathbb{R}^{n \times d_v}$:

$$Z = AV \tag{2.13}$$

This allows transformers to eliminate the sequential bottleneck and model long-term dependencies effectively across large datasets. Self-attention forms the core component of stacked encoder and decoder layers, where it is combined with additional components such as feed-forward networks and layer normalization, as illustrated in Figure 2.6. Multiple self-attention heads are typically used in parallel to allow the model to capture different types of dependencies within the same sequence.

Transformer architectures can be categorized into three types:

- **Encoder-decoder models** consist of an encoder that processes the input sequence into embeddings, and a decoder that uses these embeddings to generate the output sequence.
- **Encoder only models** [10] process the entire input sequence using bidirectional self-attention, allowing each token to attend to all others and thus capture full contextual information.
- **Decoder-only models** [36] employ masked self-attention, which restricts access to future tokens and enables auto-regressive generation by predicting the next token based solely on past observations.

These architectures are suited to different types of tasks. Encoder-only models are typically used for understanding the full input context which is useful for tasks like classification. Decoder-only models are more suited for generative tasks like text generation, where future values are predicted sequentially based on past observations. Encoder-decoder models are typically used to transform input sequences

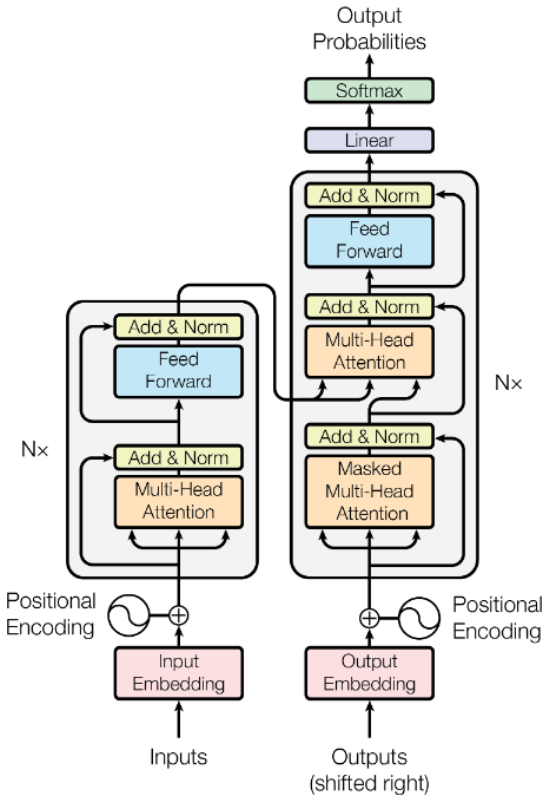


Figure 2.6: High-level architecture of the Transformer model showing encoder and decoder stacks with self-attention and feed-forward layers, adapted from Vaswani et al. [47].

into a required output sequence, useful for tasks like translation. In practice, the choice of architecture involves a trade-off between flexibility and computational efficiency.

2.4. Transformers for Time Series

2.4.1. Adapting Transformer for Time Series

Applying transformers to time series data introduced several challenges.

First, self-attention introduces computational complexity which scales quadratically with sequence length [21], since every timestep must attend to all others. Second, the standard transformer architecture does not encode temporal order [30]. Finally, time series data consists of continuous values, unlike the discrete token representations used for natural language processing, introducing additional challenges in representation and normalization [55].

Addressing these challenges requires several adaptations. Segmentation techniques like Patching [31] are often used on long sequences, where consecutive timesteps are grouped and treated as single input tokens, reducing the overall sequence length and computational cost. Positional encodings are used to inject information about the temporal order into the model. Normalization techniques like Reversible Instance Normalization (RevIN) [23] and LayerNorm [51] are used to handle distribution shifts, where input features are scaled to standardize their mean and variance, preventing internal covariate shift and improving training stability.

Together, these allow the transformer to move beyond discrete sequence modeling to continuous time series modeling.

2.4.2. Transformer-based Time Series Models

Following the success of transformers in sequence modeling, several architectures have been adapted specifically to address time series forecasting.

Informer
Informer [58] adapts the transformer architecture by introducing the *ProbSparse* attention mechanism designed to reduce computational complexity by using the dominant query and key pairs, and a generative decoder that produces entire horizon predictions in a single forward pass, eliminating the error accumulation of auto-regressive generation.

Autoformer
Autoformer [49] embeds time series decomposition directly into the transformer, separating the trend and seasonality properties across layers rather than as a preprocessing step. It also introduces an *Auto-Correlation* mechanism that efficiently identifies recurring periodic patterns, enabling the model to aggregate information across segments of points rather than treating each point independently.

Temporal Fusion Transformer
Temporal Fusion Transformer (TFT) [27] is an adaptable, interpretable model designed to handle multiple types of inputs: covariates, past observed values and future known values. It utilizes an LSTM for short-term pattern extraction and attention for long-term pattern extraction.

PatchTST
PatchTST [31] introduces segmenting of time series into fixed length patches, treating each patch as a token rather than individual time steps, significantly reducing the computation complexity. It also utilizes a channel-independent approach where each variate is processed independently through the same shared transformer backbone, reducing overfitting.

While these architectures improve forecasting performance significantly over classical statistical models, they remain largely task-specific. Each model introduces architectural decisions tailored to address specific limitations such as computational efficiency and incorporation of covariates and none can generalize across domains without task-specific adaptation or training.

This limitation motivates a shift toward more general-purpose models that aim to learn transferable representations through large-scale pre-training, reducing reliance on task-specific architectures and domain-specific tuning. As such, these transformer-based forecasting models have served as important stepping stones toward the emerging paradigm of foundation models for time series.

2.5. Foundation Models
Foundation models [57] [6] are models trained on a large diverse datasets to be able to perform a wide range of downstream tasks. They learn general representations and patterns from data via large-scale pretraining, enabling transferable knowledge that can be applied to new tasks across domains with minimal additional training. They represent a paradigm shift away from traditional machine learning, where models are trained for a single task.

Large Language Models (LLMs) are among the most successful examples of foundation models. Models like BERT (Bidirectional Encoder Representations from Transformers) [10] use a bidirectional encoder for language understanding and GPT (Generative Pretrained Transformer) [7] uses a unidirectional decoder for generation. Both are typically adapted to downstream tasks through fine-tuning, prompting or in-context learning.

The foundation model paradigm is driven by three core architectural components: **tokenization**, **pre-training**, and **transfer learning**.

2.5.1. Tokenization
Tokenization is the process of converting raw data into a sequence of discrete units called tokens, which the model can process. Modern foundation models typically use tokenization methods like Byte Pair Encoding [42] which decompose rare or unseen words into smaller meaningful units. The resulting token sequence is mapped to numerical embeddings and passed as input to the transformer. Tokenization plays a critical role as it directly determines what the model can represent: a challenge when extending foundation models beyond text to other modalities such as time series, where no natural discrete vocabulary exists.

2.5.2. Pretraining
Pretraining refers to the initial phase of training a foundation model. The models are trained on large diverse datasets using a self-supervised objective that requires no task-specific labels [28]. Though computationally intensive [19], pretraining produces rich, transferable representations that capture general structure in the data, forming the foundation for downstream adaptation.

2.5.3. Transfer Learning
Transfer learning [59] refers to adapting knowledge acquired during pretraining to new tasks or domains with minimal additional training. This is done directly via **zero-shot inference** [37], where the pretrained model is applied to a new task without any additional training or examples, relying entirely on representations learned during pretraining. This makes foundation models immediately useful across diverse domains out of the box. When some labeled data is available, **fine-tuning** [16] [7] updates the pretrained model's weights on a task-specific dataset, allowing the model to specialize its representations while retaining general knowledge acquired during pretraining. Fine-tuning typically yields stronger performance than zero-shot inference, but required labeled examples and introduces additional computational cost.

2.6. Time Series Foundation Models
Time Series Foundation Models (TSFMs) are a specialization of foundation models with a focus on time series data. Their objective is to learn temporal dependencies and mappings from large amounts of time series data in order to capture general patterns in time series behavior, and to transform this knowledge into contextual representations that can be used for downstream tasks such as forecasting and anomaly detection.

Since foundation models are typically applied to multi-modal data, with a strong focus on text, adapting them to time series requires the use of various techniques that extend transformer architectures for sequential data, as discussed earlier. Different TSFMs use different techniques and therefore differ significantly in their architecture. Some focus purely on univariate forecasting, while others explicitly incorporate covariates as part of their design.

The three models considered in this study—TimesFM, Chronos, and Moirai—represent different positions on this spectrum, differing in architecture, pretraining strategy, and the extent to which covariate handling is a central aspect of their design.

2.6.1. Google TimesFM
TimesFM 2.5 [9] is a 200-million-parameter decoder-only TSFM that demonstrates strong performance on univariate time-series datasets. In the univariate setting, the model relies solely on historical values of the target variable to learn trends and seasonal patterns. Multivariate time-series forecasting, however, incorporates exogenous variables, also referred to as covariates, to provide additional contextual information. TimesFM is trained exclusively in the univariate setting.

The model predicts future values by modeling the next patch as a function of previous patches. Patching refers to the process of grouping consecutive time steps into a single token, which reduces sequence length and improves efficiency. Each patch captures local temporal patterns, while the transformer models relationships between the patches. TimesFM is pretrained using a fully supervised learning approach and uses patch masking during training to reduce overfitting to specific context or forecast horizon lengths.

A key limitation of TimesFM is its lack of explicit support for covariates, primarily due to the scarcity of large-scale, high-quality multivariate pretraining data. The authors suggest incorporating exogenous features either by fitting a linear regression model on top of the TSFM outputs or by injecting covariates directly into the input representation so that the attention mechanism can attend to them. However for this study, we stick to the original implementation and evaluate TimesFM as a univariate forecaster.

2.6.2. Amazon Chronos
Chronos-2 [2] is 120 million parameter encoder-only model developed to address a core limitation of current time series foundation models: the assumption of most tasks being completely univariate.

Recognizing that most industrial settings are control driven, the authors designed Chronos-2 with native covariate support as a core objective.

While most TSFMs apply standard temporal self-attention, aggregating information across patches along the axis of a single series, Chronos applies an additional group attention layer, operating across multiple different time series, allowing the transformer to attend not just to the target, but also the covariates within a single forward pass. Both past and future known covariates are concatenated and processed through the same transformer stack, enabling the model to perform in-context learning, allowing zero-shot covariate handling capabilities. Input values are normalized and divided into fixed-length non-overlapping patches before being embedded and passed through alternating time and group attention layers. The authors acknowledge a significant data challenge: meaningful benchmarks with strong covariate dependence are rare, so training relied on a combination of existing univariate datasets and fully synthetic multivariate datasets for covariate informed tasks.

2.6.3. Salesforce Moirai

Moirai 1.1-R-Base [48] is a 91 million parameter (Base model) encoder-only model developed to address the limitations of task specific deep learning models and targeted specific challenges like handling varying dimensionality and time series with varying time steps.

Rather than assuming a fixed dimensionality, the authors propose Any-variate Attention, a technique which first flattens all variates, including covariates, into a single sequence and then applies full self-attention. Cross-frequency learning is handled through multiple patch size projection layers with varying sizes: larger patches for high-frequency data and smaller patches for low-frequency data.

While this represents an advancement over models like TimesFM which are limited to univariate inputs, Moirai makes no architectural distinction between target variates and covariates: both are treated as interchangeable elements of a flattened sequence. Covariate support is therefore an emergent property rather than a central architectural concern, which contrasts models like Chronos.

3

Scientific Paper

Beyond Task-Specific Models: Zero-shot Time Series Foundation Models for Geothermal Systems Monitoring

Zeryab Alam
EEMCS,
Delft University of Technology
The Netherlands

Pejman Shoeibi Omrani
Hydrology and Reservoir Engineering,
Geological Survey of the Netherlands, TNO
The Netherlands

Kubilay Atasu
Department of Software Technology,
Delft University of Technology
The Netherlands

Abstract

Geothermal energy plays an increasingly important role in decarbonizing heating, cooling, and power production. As geothermal systems operate under extreme temperatures, pressures, and subsurface uncertainties, maintaining reliable operation is critical to sustaining a continuous energy supply and reducing the total cost of ownership. Ensuring the safe and efficient operation of geothermal plants therefore requires continuous monitoring of complex, multivariate sensor streams to detect equipment degradation and anticipate operational failures before they occur. This often relies on separate specialized physics-based and machine learning models for each task, with sparse labels and inter-site variability limiting generalization. In this work, we explore the application of state-of-the-art Time Series Foundation Models (TSFMs) as a unified alternative for both forecasting and anomaly detection in geothermal operations. We present a geothermal-specific benchmark for time series modeling and conduct a systematic comparative evaluation of conventional machine and deep learning baselines against pretrained TSFMs under zero-shot conditions. The results demonstrate that, in forecasting tasks, covariate-aware TSFMs, particularly Chronos, consistently outperform all trained baselines, achieving 22–35% lower RMSE across horizons. For anomaly detection, we evaluate multiple detection strategies and find that performance is influenced more strongly by the choice of detection strategy and the availability of labeled fault data than by forecasting accuracy alone, with TSFM embeddings consistently encoding system information and enabling effective anomaly detection under labeled conditions. These findings establish TSFMs as a promising foundation for intelligent condition monitoring in geothermal and broader industrial time series applications, while highlighting the importance of explicit covariate modeling for this class of systems.

1 Introduction

The transition towards renewable energy sources has made geothermal energy an increasingly important sustainable alternative to hydrocarbons (IEA, 2024). Unlike intermittent solar or wind resources, geothermal energy provides reliable thermal energy derived from the Earth's natural heat. However, its long-term viability is strongly influenced by subsurface conditions, where harsh thermal, mechanical, and chemical conditions lead to persistent operational challenges such as system degradation (Omrani & de With, 2025). Consequently, the primary challenge in modern geothermal operations is not merely the extraction of heat, but the monitoring and mitigation of systemic degradation.

Geothermal energy is harvested at varying depths using different types of systems, such as closed-loop and open-loop configurations, for applications including power generation and direct-use heating. These applications can be broadly categorized into three main types. The first involves ground-source heat pumps (Kumar & Alam, 2025), that take advantage of the Earth's stable subsurface temperature to provide efficient heating for buildings. The second category is geothermal power generation (Zarrouk & Moon, 2014), where high temperature geothermal resources are utilized to produce electricity. The third category, and the focus of this work, is geothermal direct use (J. W. Lund & Toth, 2021) (J. Lund, 2006). In these systems, geothermal fluids with temperatures ranging from 30°C to 150°C are extracted and passed through heat exchangers to transfer thermal energy to secondary systems, such as district or building heating networks (Brown et al., 2022). The cooled geothermal fluid is then reinjected into the

reservoir.

The operational integrity of this cycle depends on three tightly interconnected subsystems (Whole Building Design Guide (WBDG), n.d.) (Purwaningsih & Abdurrahman, 2016), as shown in Figure 1. The production facility consists of wells drilled to bring the hot geothermal fluids, referred to as brine, to the surface using an ESP, an electrical submersible pump. Once at the surface, the fluid flows through the mechanical system, which includes the piping, separator, filters and the heat exchanger (HEX). After the thermal energy has been extracted, the brine is then injected back into the reservoir using the injection facility, which consists of an injector/booster pump and the injector well.

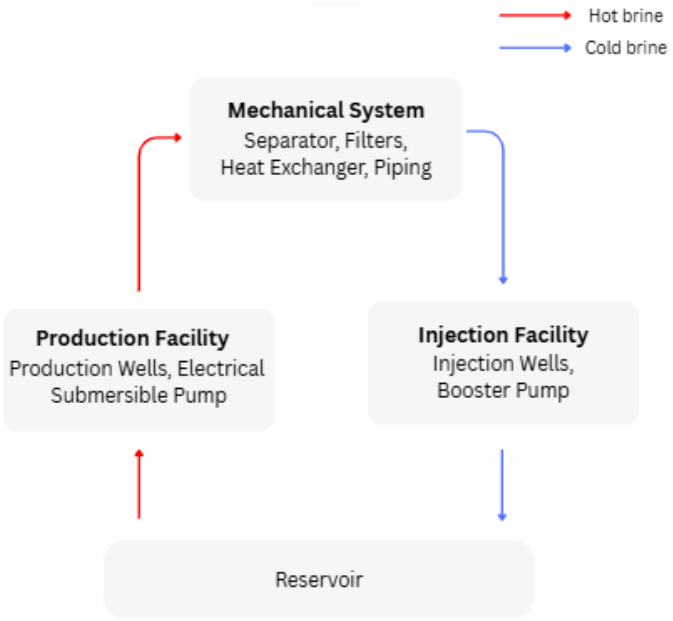


Figure 1: Simplified illustration of the direct-use geothermal system, where hot brine is extracted via a production facility and thermal energy is extracted via the mechanical system, and cool brine is reinjected back via the injection facility.

Due to this interconnected nature, the systems are complex to operate and require careful monitoring and maintenance to detect and service degradations in various components (Axelsson & Stefánsson, 2003). The most common forms of anomalies and degradation in geothermal systems include scaling in wells and pipelines, fouling, corrosion, and complexities in pumping systems. Scaling (Pambudi et al., 2015) is the unwanted deposition of minerals present in the brine inside the pipes and components. Fouling (Ogbonnaya & Ajayi, 2017) (Penot et al., 2023) is more broader than scaling, but also covers organic matter. Corrosion (Nogara & Zarrouk, 2018) is the degradation caused by chemical reactions between the high temperature geothermal brine and the system components. Pumping systems (Fakher et al., 2021) undergo constant stress and are more prone to reduced efficiency and sudden failures.

Degradation in these components, even if not immediately significant, can progressively impact system performance and if left unaddressed, may lead to severe operational interruptions or complete shutdowns, resulting in significant economic and operational consequences (Shannon, 1975). Hence, it is necessary to ensure healthy operation of equipment to promote safety and reduce downtime.

Following the operational challenges associated with geothermal system degradation, machine learning and data-driven approaches are increasingly being adopted to support predictive maintenance, condition monitoring, and anomaly detection in geothermal operations (Omrani et al., 2025a) (Omrani et al., 2025b). These methods aim to identify degradation patterns such as scaling, fouling, and pump failures before they develop into severe operational disruptions. Recent studies have explored the applicability of such techniques across different degradation mechanisms. Al Harrasi et al. (2025) demonstrated the use of ensemble models for improved prediction of scaling formation. For heat exchanger systems, Villa and Brusamarello (2025) investigated machine learning approaches for fouling detection and reported that memory-based models such as Long Short-Term Memory (LSTM) networks were particularly effective due to their ability to capture temporal dynamics and incorporate covariates.

Extensive research has also been conducted on fault diagnosis and predictive maintenance of ESPs, which are widely used in geothermal and broader energy production systems (Omrani et al., 2021) (Yang et al., 2022). For instance, Wei et al. (2024) determined that data-driven approaches stick to using limited characteristics to determine faults. They established that adding physical constraints to an LSTM model greatly decreases error in the predictions of the multiple different ESP faults, while Abdalla et al. (2022) showcased an XGBoost model's predictive power on detecting faults and Peng et al. (2021) used a Principal Component Analysis (PCA) based method for the same purpose.

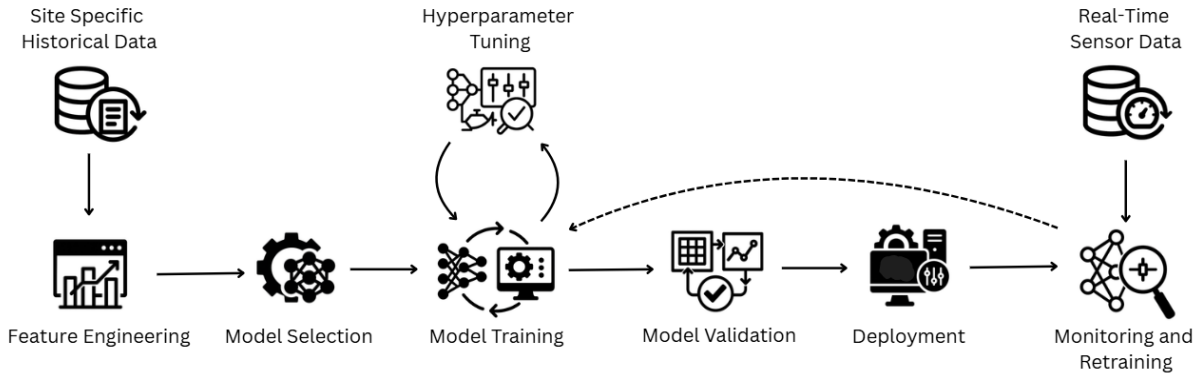


Figure 2: Illustration of ML workflows that are standard in the industrial domains. Involves data collection, feature engineering, model selection, model training, hyperparameter tuning, validation and finally deployment. Models are also monitored and retrained periodically to adapt them to regime shifts.

Despite these advances, these approaches are typically developed for specific wells or operating conditions, and often exhibit limited generalization when applied to unseen sites due to differences in geological properties and operating regimes. As a result, models must frequently be retrained for each deployment scenario. A typical machine learning workflow (Wirth & Hipp, 2000) for training and deploying models to perform monitoring in industrial systems is shown in Figure 2. This is of course, simplified to an extent, as proper maintenance and deployment of conventional models involve extensive orchestration and management (Kreuzberger et al., 2023). This process is often computationally expensive and resource-intensive for industrial operators. In addition, separate models are commonly required for different downstream tasks such as forecasting, fault detection and classification, leading to increased development and model maintenance overhead. A conventional model trained to forecast a specific target is tightly coupled to that task and cannot be repurposed for other objectives, hence being inherently task-specific. Furthermore, the limited availability of labeled failure data in geothermal settings remains a major bottleneck for supervised approaches (Nunes et al., 2023). Faults are rare events and the data is often siloed within companies, causing models to overfit on limited examples and fail to generalize across sites (Yang et al., 2022).

These limitations point toward a fundamental requirement: models that can generalize across sites and tasks without depending on large amounts of labeled site-specific data. Foundation Models (Bommasani et al., 2021) represent a paradigm shift in this regard, taking the principle of transfer learning — adapting representations from one context to new settings with minimal retraining (Zhuang et al., 2019) — and applying it at a much larger scale. Pretrained on large and diverse datasets, they learn general representations and patterns that can be adapted to a wide range of downstream tasks without retraining from scratch. Their architecture further supports multiple modalities such as text, images, and time series, making them versatile starting points for a broad class of industrial problems.

Time Series Foundation Models (TSFMs) (Liang et al., 2024) are a specialization of FMs that focus on learning representations from time-series data. Their objective is to capture temporal dependencies and extract meaningful information to perform tasks such as forecasting, anomaly detection, and event modeling. Geothermal operations are inherently data-rich, continuously generating multivariate sensor streams that record system behavior over time. This makes TSFMs particularly well-suited to address the challenges described above, as condition monitoring in these systems involves detecting deviations, anomalies, and degradation patterns within these signals (Kottapalli et al., 2025). Beyond their predictive capabilities, the contextual embeddings — internal representations of the input time series — produced by TSFMs may encode meaningful information about system state, potentially capturing degradation

trends or anomalous operating conditions even without task-specific supervision. They can be deployed under different inference strategies depending on the availability of labeled data. In zero-shot inference, the pretrained model is applied directly to a new task without any additional training, relying entirely on representations learned during pretraining. In few-shot inference, a small number of labeled examples are provided as context to guide predictions (Iwata & Kumagai, 2020). Fine-tuning, by contrast, involves continuing to train the model on task-specific data, allowing it to adapt its learned representations to the target domain (Das et al., 2024a). While fine-tuning can produce meaningful performance improvements, it requires sufficient labeled data and introduces computational overhead (Pratap et al., 2025). Training TSFMs from scratch is even more demanding, requiring large scale compute infrastructure and vast datasets (Sun et al., 2024), resources that are rarely available in industrial deployments. This motivates evaluating TSFMs primarily in the zero-shot setting, which represents the most practically accessible deployment scenario for data-scarce environments such as geothermal operations.

In the zero-shot setting, TSFMs bypass the requirement for extensive site-specific historical data entirely, directly addressing the data scarcity and generalization limitations that constrain traditional approaches. Rather than training a separate model per site or per task, a single pretrained TSFM can be applied across varying operating conditions and downstream objectives, reducing deployment overhead while improving robustness. Replacing conventional models in Figure 2, a zero-shot TSFM offers the potential to reduce the overhead associated with task-specific data collection, feature engineering, and model tuning. This work investigates precisely this scenario: we evaluate existing pretrained TSFMs on geothermal operational data, assessing how well their zero-shot capabilities transfer to a setting where system behavior is driven primarily by operational covariates rather than recurring temporal patterns.

Recent research (see Table 1) highlights the evolving role of TSFMs across diverse industries. Although their industrial adoption remains in its early stages, several sectors have begun exploring their potential across a range of tasks. A notable trend is that most successful demonstrations of zero-shot capabilities have been reported in predominantly univariate settings, while applications involving external covariates have largely relied on fine-tuning. This distinction is particularly evident in domains such as energy systems and finance, where covariate information plays a central role and zero-shot performance remains limited. To our knowledge, this work is the first to evaluate TSFMs in a purely zero-shot setting on a covariate-driven geothermal energy system, with strong zero-shot performance representing a key contribution of this study.

Study	**Domain**	**Input Type**	**Training Type**	**Zero-shot Performance**
Shetty et al. (2025)	Industrial Assets	Covariate-aware	Pretraining + fine-tuning	Non-applicable
Park et al. (2025)	Energy Systems	Univariate	Fine-tuning	Suboptimal (fine-tuning essential)
Meyer et al. (2025)	Smart Grids	Univariate	Zero-shot	Strong
Marconi (2025)	Finance	Covariate-aware	Fine-tuning	Suboptimal (fine-tuning required)
This work	Geothermal Energy Systems	Covariate-aware	**Zero-shot**	**Strong**

Table 1: Overview of recent TSFM studies, focusing on data characteristics, training regimes, and zero-shot applicability across domains.

Applying TSFMs in the energy systems operation introduces an additional layer of complexity. Geothermal systems are subject to frequent operator interventions and control actions that strongly influence system behavior, introducing covariate complexity that general-purpose TSFMs, typically pretrained on univariate series (Qin et al., 2025), are not inherently designed to handle covariates. This raises a practical challenge of how to effectively integrate control-related covariates into TSFMs. Crucially, our work highlights that recent advancements in covariate integration frameworks can achieve strong performance out-of-the-box, potentially mitigating the need for resource-intensive fine-tuning. To investigate

this, we evaluate TSFMs under different input configurations, comparing univariate and covariate-aware setups to assess the impact of covariate incorporation on model performance.

Benchmarking TSFMs on this class of problems requires datasets that reflect covariate-driven dynamics; however, existing time-series forecasting benchmarks provide limited insight, as they often lack meaningful covariates (Shchur et al., 2025). As a result, many recently proposed models are evaluated primarily in univariate or weakly multivariate settings (Arango et al., 2025), where performance is driven largely by seasonality or historical trends, rather than external control signals. This gap highlights the need for benchmark frameworks tailored to covariate-driven settings.

This paper presents the first systematic evaluation of TSFMs in the geothermal energy domain, focusing on their applicability, benefits, and limitations for condition monitoring and anomaly detection in geothermal plant operations. To enable a controlled and meaningful evaluation, we introduce a benchmarking framework that simulates key aspects of control-driven industrial geothermal systems, including operational interventions and covariate dynamics characteristic of real plant behavior. Using this framework, we benchmark multiple TSFMs to assess their performance in forecasting and anomaly detection tasks. Specifically, we address the following research questions:

1. **How well do state-of-the-art pretrained Time Series Foundation Models perform for condition monitoring and anomaly detection in geothermal operations under zero-shot settings?**
2. **What are the benefits and limitations of using TSFMs compared to traditional machine learning models in this context?**

Our results show that TSFMs outperform all trained baselines across forecasting tasks, with larger gains at longer horizons where task-specific models degrade most. Their embeddings also serve as effective representations of system dynamics, enabling strong and consistent anomaly detection across all evaluated fault types when used with a supervised classifier. Overall, TSFMs show strong potential for industrial condition monitoring.

The paper is structured as follows. The Methodology section introduces the overall pipeline, covering the selected TSFMs, forecasting setup, anomaly detection approach, and evaluation metrics. The Case Study section describes the three degradation events and how they inform the construction of the benchmark datasets. The Results and Evaluation section presents the forecasting and anomaly detection outcomes alongside an analysis of findings. The Discussion section examines key insights, limitations, and opportunities for improvement, and the paper concludes with a summary of contributions and directions for future work.

2 Methodology

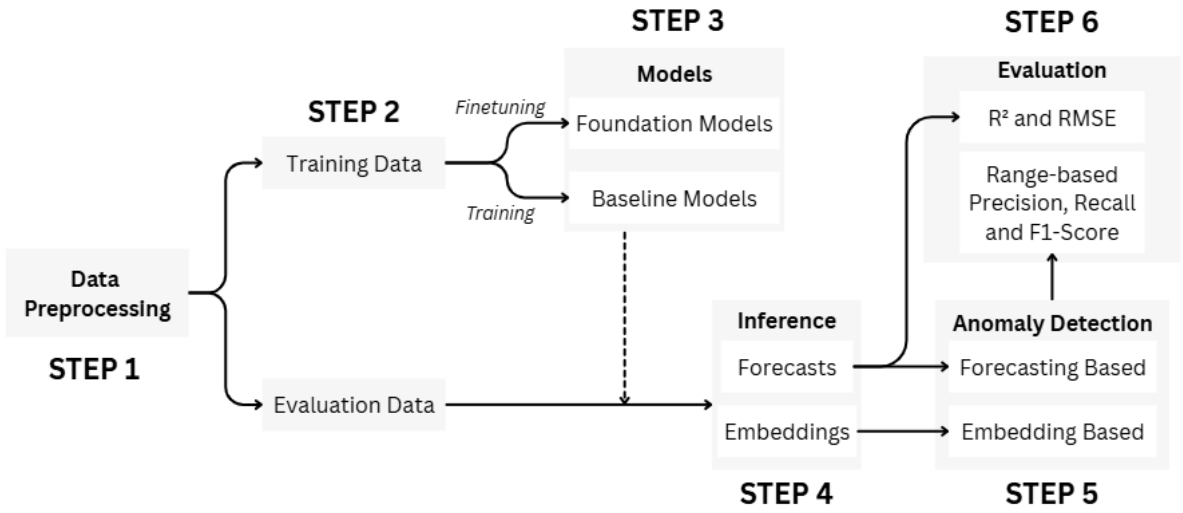


Figure 3: The overall framework showcases the complete pipeline: from preprocessing to model inference and downstream evaluation. Solid lines indicate data flow, while dashed lines represent model application to evaluation data for inference and downstream application.

The overall framework schematic is displayed in Figure 3. The pipeline is designed to systematically evaluate the applicability of TSFMs by providing a standardized workflow for benchmarking their performance against conventional baseline models across forecasting and anomaly detection tasks.

The input to the pipeline is multivariate time-series data representative of a geothermal direct-use system, comprising sensor readings including pressures, temperatures, flow rates, pump frequencies, and power output, among many others. This data serves as the basis for all downstream tasks: covariate-informed forecasting for condition monitoring, and anomaly detection for degradation identification. Each step of the pipeline is described below.

Step 1: Data Preprocessing

Raw sensor data is first preprocessed to ensure it is physically consistent and suitable for model ingestion. Missing values are imputed using column-wise means, and out-of-range or physically invalid readings are removed to filter sensor malfunctions. Since the dataset is synthetically generated, ground-truth anomaly labels are available and used directly for evaluation without additional labeling steps. Finally, feature scaling is applied where required: deep learning models are trained on min-max normalized inputs, while foundation models perform normalization internally and tree-based models require no scaling.

Step 2: Model Ready Data

The preprocessed data is partitioned into training and evaluation splits. Since this study focuses on zero-shot evaluation of TSFMs, the foundation models require no training data and are applied directly to the evaluation set. However, the framework retains the training split as an essential component for two reasons: first, the baseline models are trained on this data; second, fine-tuning of TSFMs on domain-specific data remains an open and natural direction for future work, and the framework is designed to support this without modification. All models, whether trained from scratch or applied zero-shot, are subsequently evaluated on the held-out evaluation data in the steps that follow.

Step 3: Models

The model block is designed to be modular, allowing any forecasting model to be slotted into the pipeline. Downstream processes such as inference, forecasting-based anomaly detection, and evaluation are designed to operate on models that produce sequential forecasts, with model-specific adaptations where necessary.

Embedding-based anomaly detection additionally requires the model to expose its internal latent representations. The block is divided into two components: the foundation models under evaluation, and the conventional baseline models used for comparison.

Step 3A: Foundation Models

Three state-of-the-art TSFMs are evaluated in this study, selected to represent three distinct architectural paradigms: dedicated covariate attention, unified variate flattening, and univariate-only decoding. All three are evaluated in a purely zero-shot setting, with no finetuning or task-specific adaptation applied.

- **Amazon Chronos-2** (Ansari et al., 2025): a 120M encoder-only model with a dedicated cross attention mechanism that natively incorporates covariates within the attention computation, enabling zero-shot covariate-aware forecasting.
- **Salesforce Moirai 1.1-R-Base** (Woo et al., 2024): a 91M encoder-only model that supports arbitrary numbers of variates by flattening all inputs, including covariates and targets, into a single sequence before applying self-attention.
- **Google TimesFM 2.5** (Das et al., 2024b): a 200M decoder-only model operating exclusively in the univariate setting, evaluated here without covariate support as per its original implementation.

Step 3B: Baseline Models

Several conventional models serve as baselines to interpret TSFM performance. All baseline models are trained on the training data, in contrast to the TSFMs which operate zero-shot. Model configurations are provided for reproducibility and fair comparison.

- **Random Forest** (Breiman, 2001): a quantile regression forest with 200 estimators, implemented using `scikit-learn` (Pedregosa et al., 2011). Operates as a pointwise regressor without temporal structure, and is therefore evaluated only at a fixed horizon.
- **LSTM** (Hochreiter & Schmidhuber, 1997) (Ben Aoun et al., 2026): an encoder-decoder recurrent network implemented in `scikit-learn`. A 128-neuron encoder LSTM summarises past observations and target values into a context vector, which is concatenated with known future covariates and passed to a 128-neuron decoder LSTM with dropout. A time-distributed output layer produces three quantiles per forecast step, enabling probabilistic multi-horizon forecasting.
- **TiDE** (Das et al., 2024c): a Time-series Dense Encoder implemented using the `Darts` library (Herzen et al., 2022). Configured with 2 encoder and 2 decoder layers, a 256-neuron hidden layer, and a 64-neuron temporal decoder with layer normalisation and dropout regularisation. Produces multi-horizon probabilistic forecasts via quantile regression.
- **Naive Model**: a simple persistence baseline that repeats the last observed value across the entire forecast horizon. This serves as a reference to quantify task difficulty. It is used exclusively for forecasting and is not applied to anomaly detection tasks.

Step 4: Inference

During inference, models are provided with a fixed-length context window of past observations, consisting of historical features and target values, together with known future features, and are tasked with predicting future target values over a predefined horizon. In our setting, the features correspond to operational control variables used by plant operators to regulate system behavior. Figure 4 illustrates this process for a single inference window.

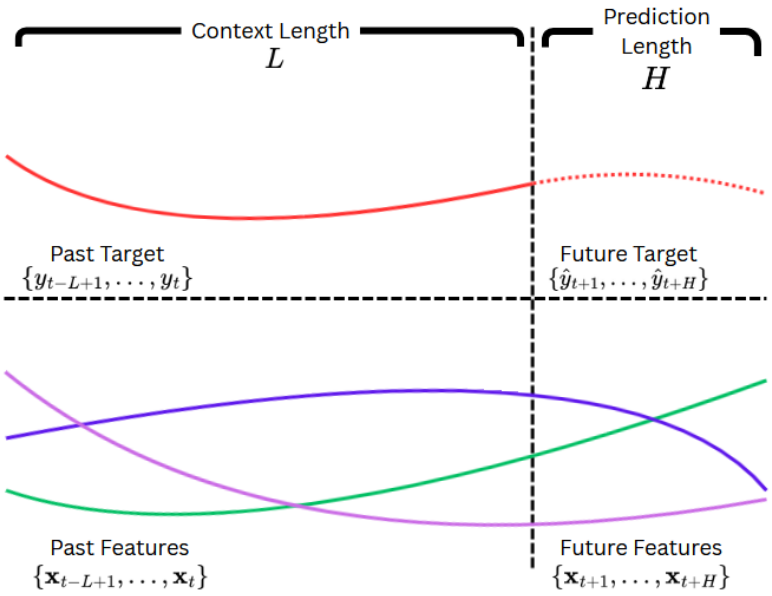


Figure 4: Illustration of the time-series forecasting task during inference using a sliding window approach. The model ingests a context of past targets and features alongside known future features to produce forecasts over the defined horizon.

Formally, let $\mathbf{x}_t \in \mathbb{R}^d$ denote the multivariate feature vector at time t, where d is the number of input variables, and let $y_t \in \mathbb{R}$ denote the scalar target variable. Given a context length L and prediction horizon H, the model receives the historical context $\{(\mathbf{x}_{t-L+1}, y_{t-L+1}), \ldots, (\mathbf{x}_t, y_t)\}$ together with known future features $\{\mathbf{x}_{t+1}, \ldots, \mathbf{x}_{t+H}\}$, and predicts the future target sequence $\{\hat{y}_{t+1}, \ldots, \hat{y}_{t+H}\}$. Specifically, for each future timestep, the model outputs a lower prediction bound, point forecast, and upper prediction bound, $\{(\hat{y}^{\text{low}}_{t+h}, \hat{y}_{t+h}, \hat{y}^{\text{up}}_{t+h})\}_{h=1}^{H}$, where $\hat{y}^{\text{low}}_{t+h}$ and $\hat{y}^{\text{up}}_{t+h}$ correspond to the empirical 2.5% and 97.5% quantiles, respectively, yielding a 95% prediction interval that captures predictive uncertainty in addition to the expected target trajectory. For all models, the quantile levels are defined as $(0.025, 0.975)$. The only exception is TimesFM, for which $(0.1, 0.9)$ is used instead, due to its quantile head being restricted to fixed 10% intervals.

Inference follows a sliding window procedure. For a time series of length T, windows are generated by iteratively shifting the starting position using a stride S, such that the i-th window begins at time $t_i = i \cdot S$. In this study, the stride is set equal to the prediction horizon ($S = H$), producing non-overlapping forecast segments.

Step 5: Anomaly Detection

Anomaly detection aims to identify abnormal system behavior from multivariate time-series observations. Formally, given a sequence of input features $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_T\}$ and corresponding target observations $\mathbf{Y} = \{y_1, \ldots, y_T\}$, the objective is to assign an anomaly label $a_t \in \{0, 1\}$ to each timestep t, where $a_t = 1$ indicates anomalous behavior and $a_t = 0$ denotes normal operation.

As is the case in many industrial equipment systems, geothermal plant environments exhibit anomalies that are typically not expressed as isolated point events but rather as gradual and persistent degradations in system performance. Equipment degradation often manifests through reduced operational effectiveness that progressively develops over time and persists until maintenance is performed or components are replaced. As such, anomalous behavior is commonly characterized by extended temporal intervals rather than abrupt spikes or drops in sensor values (Bondad & Redoña, 2026).

Following the anomaly detection taxonomy presented by Boniol et al. (2024), the methods employed in this study span multiple methodological categories, including prediction-based, density-based, and classification-based approaches. However, rather than organizing methods solely according to their anomaly detection mechanism, we group them based on the type of representation produced by the model during inference (Darban et al., 2024).

Specifically, we distinguish between two categories: *forecasting-based anomaly detection* and *embedding-based anomaly detection*. The former utilizes the model's predicted future target values and associated uncertainty estimates to detect deviations between expected and observed system behavior. The latter instead operates on latent embedding representations generated from the historical context window, treating these embeddings as compact summaries of system state from which anomalous operating regimes can be identified. This distinction enables the separation of short-term predictive deviations from longer-term shifts in system dynamics that may not be directly reflected in forecasting error alone (Delibasoglu & Heintz, 2024).

In total, we evaluate four anomaly detection strategies. Forecasting based methods include a novelty detection approach and a supervised Random Forest classifier, representing unsupervised and supervised uses of forecasting outputs, respectively. Embedding based methods include a density-based Isolation Forest and a supervised Multi Layer Perceptron (MLP) classifier, enabling the evaluation of both unsupervised and supervised anomaly detection on learned latent representations.

Step 5A: Forecasting Based Anomaly Detection

Step 5A.1 Novelty Detection

Forecasting model uncertainty can serve as an indicator of anomalous behavior (Ovadia et al., 2019): model uncertainty spikes during anomalous regimes due to deviations from expected system dynamics given the preceding context. Leveraging this, we employ an empirical coverage approach (Gneiting & Raftery, 2007)

Let $y_{t:t+H-1} = \{y_t, \ldots, y_{t+H-1}\}$ be the observed values over a forecasting horizon $H = 24$, and let $\hat{q}_\alpha(t+h)$ denote the predicted quantile at level $\alpha \in (0,1)$ for horizon step h. The empirical coverage (EC) of a prediction interval $[\hat{q}_{\alpha_l}, \hat{q}_{\alpha_u}]$ over the window starting at time t is defined as

$$\mathrm{EC}(t) = \frac{1}{H} \sum_{h=0}^{H-1} I\left(\hat{q}_{\alpha_l}(t+h) \leq y_{t+h} \leq \hat{q}_{\alpha_u}(t+h)\right), \tag{1}$$

where $I(\cdot)$ is the indicator function. For a nominal $(1-\alpha)$ prediction interval with

$$\alpha_l = \frac{\alpha}{2}, \qquad \alpha_u = 1 - \frac{\alpha}{2}, \tag{2}$$

a window is flagged as anomalous if

$$\mathrm{EC}(t) < 1 - \alpha. \tag{3}$$

In this work, $\alpha = 0.05$, corresponding to a 95% prediction interval. Accounting for cases where the model exhibits very high uncertainty, resulting in excessively wide prediction intervals, an additional condition is used. The mean interval width is defined as

$$w(t) = \frac{1}{H} \sum_{h=0}^{H-1} \left(\hat{q}_{\alpha_u}(t+h) - \hat{q}_{\alpha_l}(t+h)\right). \tag{4}$$

Let σ_{clean} denote the standard deviation of the target variable computed on clean training data. A window is also classified as anomalous if

$$w(t) > 3\,\sigma_{\text{clean}}. \tag{5}$$

For TimesFM, which produces quantiles only at multiples of 10%, the $[0.1, 0.9]$ prediction interval is used, and a window is flagged as anomalous if

$$\mathrm{EC}(t) < 0.8. \tag{6}$$

These thresholds enable consistent model comparison to demonstrate the methodology; optimizing threshold selection is left for future work.

Step 5A.2: Classifier Random Forest

A supervised classification model is trained on a subset of labeled anomalous data and applied to the outputs of the forecasting model to provide early warning signals for operators. In this implementation, the classifier $g(\cdot)$ is instantiated as a Random Forest model (Breiman, 2001), chosen for its robustness and ability to capture non-linear boundaries. Given labeled training data $\mathcal{D}_{\text{cls}} = \{(x_t, s_t, y_t)\}$, where x_t

represents control inputs (features), s_t the target signal, and $y_t \in \{0, 1\}$ the anomaly label, the classifier is trained to map feature–signal pairs to anomaly labels: $g : (x_t, s_t) \mapsto y_t$.

At inference time, the forecasting model produces a H-step-ahead prediction $\hat{\mathbf{s}}_{t:t+H-1}$ given observation up to time t, which is then combined with the corresponding feature inputs, x_t, and passed to the classifier to obtain the predicted label $\hat{y}_t = g(x_t, \hat{\mathbf{s}}_{t:t+H-1})$. An anomaly is flagged whenever $\hat{y}_t = 1$, enabling detection on predicted future behavior rather than observed values. Performance consequently depends on both the quality of labeled fault data and the forecasting model's ability to capture the system's temporal dynamics.

Step 5B: Embedding Based Anomaly Detection

Rather than operating on forecast outputs, embedding based methods exploit the latent representations produced by the model during inference. Models that generate such representations produce a compact embedding for each context window, summarizing the historical system state, such that each sliding window yields both a forecast and an associated context embedding. As detailed in Table 2, these raw internal representations often consist of multiple data patches across time and variables, which must be condensed into a fixed-length feature vector. To achieve this, we employ pooling (Gholamalinezhad & Khosravi, 2020), specifically average pooling across temporal patches, which aggregates information by computing the mean representation within each patch. This is followed by flattening, which reshapes the resulting multi-dimensional tensor into a one-dimensional feature vector for downstream processing.

Model	Raw Embedding Shape	Pooling Strategy	Final Dimension
LSTM	(128,)	None	(128,)
TiDE	(256,)	None	(256,)
Chronos	(6, 13, 768)	Pooling + Flattening	(4608,)
Moirai	(144, 768)	Pooling	(768,)
TimesFM	(6, 1280)	Pooling	(1280,)

Table 2: Embedding dimensions and pooling strategies. For LSTM and TiDE, dimensions reflect the selected encoder layer size. For TSFMs, the first dimension represents temporal patches (144 for Moirai; 6 for TimesFM). For Chronos, the shape $(6, 13, 768)$ denotes 5 covariates plus 1 target variable, each with a patch size of 13. Pooling is applied to condense temporal patches into a fixed-length vector for anomaly detection.

Step 5B.1: Isolation Forest

An Isolation Forest (Liu et al., 2009) is fitted on embeddings extracted from a known clean period of operation: the first year of data consisting of 360 embedding vectors. As an unsupervised anomaly detection method, it does not require labeled anomalous examples during training and instead learns the structure of normal operating behavior directly from healthy data. It operates on the principle that anomalous observations are structurally easier to isolate than normal ones: anomalies require fewer random binary splits to be separated from the rest of the data, and thus have shorter average path lengths in the ensemble of isolation trees. Since embedding dimensionality varies across models, hyperparameter tuning is performed via grid search over the number of estimators, maximum sample size, feature fraction, and bootstrap strategy as mentioned in Table 10 (Appendix C), rather than manually selecting parameters for each model individually. Hyperparameter selection maximizes $\mu - \sigma$ of the training decision function scores, favoring configurations that assign consistently high normality scores to the clean training data. The fitted model is then evaluated on embeddings extracted from the remaining period of the dataset.

Step 5B.2: Classifier MLP

Similar to the Random Forest classifier, a supervised MLP classifier is trained on labeled embeddings to distinguish between normal and anomalous operating regimes. Since model embeddings can be high-dimensional, a feedforward neural network is chosen over simpler classifiers for its capacity to learn non-linear decision boundaries in high-dimensional spaces (Hornik et al., 1989).

As with Isolation Forest, the MLP dynamically adapts its input dimensionality to match the embedding dimension d of the evaluated model. Because these embeddings vary significantly across architectures, the classification network must adjust to optimally leverage each feature space; thus, rather

than using a static configuration, the MLP undergoes hyperparameter tuning via grid search over the configurations detailed in Table 10 (Appendix C). Implemented using `scikit-learn`, the optimized MLP maps each embedding $\mathbf{e}_i$ to a binary anomaly label: $\hat{y}_i = g(\mathbf{e}_i) \in \{0, 1\}$, where $\hat{y}_i = 1$ indicates anomalous behavior. As with the forecasting based classifier, performance depends on the separability of the embedding space and the quality of the labeled training data.

Step 6: Evaluation

Forecasting is evaluated using standard accuracy metrics:

- R^2: Coefficient of Determination, measuring the proportion of variance in the target explained by the model. Given true values $y_1, \ldots, y_N$, predictions $\hat{y}_1, \ldots, \hat{y}_N$, and mean $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i$, it is computed as
$$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}. \tag{7}$$
- $RMSE$: Root Mean Squared Error, measuring the average magnitude of the prediction error, defined as
$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}. \tag{8}$$

Anomaly detection is evaluated using range-based precision and recall (Lee et al., 2018) to account for the temporal characteristics of geothermal degradation events. In this work, we use the implementation provided by Ryohei Izawa (2021). The `cardinality` parameter is set to "reciprocal", which penalizes fragmentation and encourages models to produce continuous flags. The `bias` parameter is set to "front" to prioritize the early portion of an anomalous segment, which is critical for early warning of onset degradation. Precision is computed with a threshold of 0.0, counting any detection within the anomaly range as a true positive, while recall uses a threshold of 0.6 to require moderate coverage of the anomaly for a positive detection. The F1 score is then computed as
$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{9}$$

Precision, recall and F1-scores are reported for each task. Parameters are fixed to standardize evaluation, prioritizing model comparison over hyperparameter optimization.

Implementation Details

All experiments were conducted on a Microsoft Azure cloud virtual machine with configuration *Standard_NC6s_v3*, equipped with 6 CPU cores, 112 GB RAM, 736 GB disk storage, and a single NVIDIA Tesla V100 GPU.

Experiments were implemented in Python using Jupyter notebooks. A variety of libraries were used depending on the model, including `azure-ai-ml`, `numpy`, `pandas`, `scikit-learn`, `sklearn-quantile`, `matplotlib`, and deep learning frameworks such as `TensorFlow` and `PyTorch`. Model-specific libraries included `chronos-forecasting`, `uni2ts`, `darts`, and `pytorch-lightning`.

Due to compatibility constraints, dataset generation and experiments involving the TimesFM model were executed on a local machine with an Intel Core i5-1235U processor, 16 GB RAM, and integrated Intel Iris Xe graphics.

3 Case Study and Datasets

3.1 System Overview

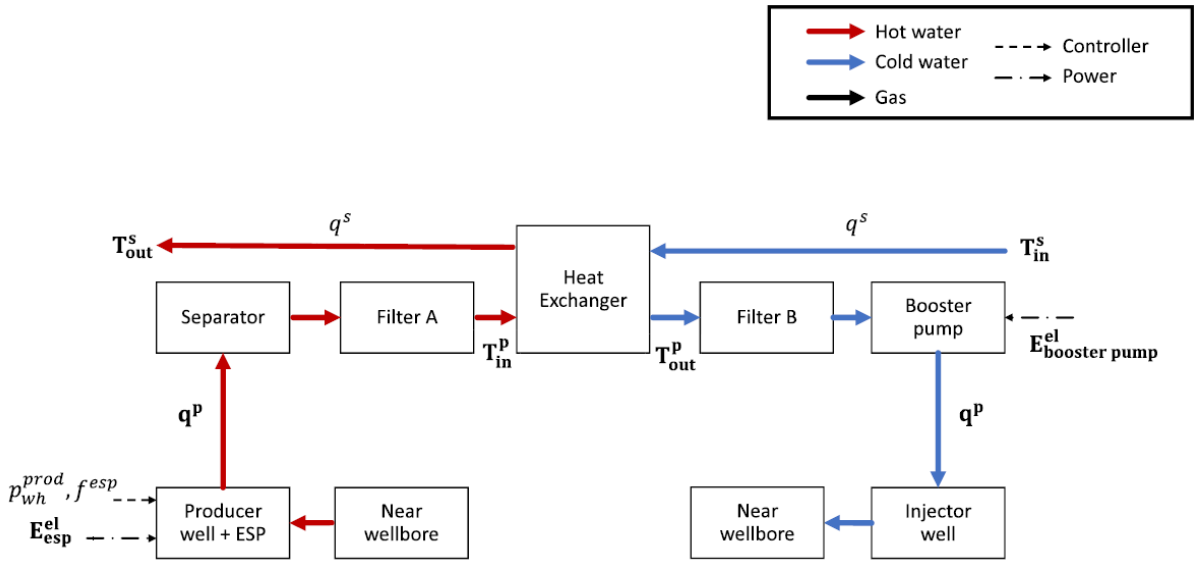


Figure 5: Overview of the geothermal direct use system, showcasing the flow of geothermal brine from the production well through separator, filters, and heat exchanger, along with associated control and power inputs.

The core components of the geothermal plant for direct-use heating application are showcased in Figure 5. The scope of this study is limited to hydrothermal deep geothermal system intended for direct use heating applications, where high temperature brine is produced from the subsurface and used directly as a heat source without conversion to electricity. Geothermal brine is extracted from the production well using an Electrical Submersible Pump (ESP). The fluid then flows sequentially through a separator, filter A, a heat exchanger, filter B, and is finally reinjected into the reservoir via the injection well.

The system operates under tightly controlled conditions, where various different components influence overall thermodynamic and hydraulic behavior. This makes it a suitable testing ground for evaluating forecasting and anomaly detection methods in complex industrial systems.

3.2 Task Definition

Model performance is evaluated through two main tasks: covariate-informed forecasting and anomaly detection via forecasting-based or embedding-based approaches.

While the broader framework forecasts multiple system variables, a more thorough analysis is performed for the case study of the heat exchanger secondary side outlet temperature. This variable is selected as the primary benchmarking target because it represents a critical operational state with tight, non-linear dependencies on input control and state variables (including flow rates, pressures, and pump behavior).

The anomaly detection tasks target three degradation events, each chosen to represent a different fault mechanism:

- **Heat Exchanger Fouling:** Represented as a decline in the system's thermal efficiency.
- **Electrical Submersible Pump Degradation:** Represented as a decline in the pump's generated head.
- **Filter Clogging:** Represented as an increase in the pressure drop across the filter.

3.3 Simulation Environment

To generate realistic datasets, we adapt the GEMINI Geothermal Digital Twin (Omrani et al., 2026) (Omrani et al., n.d.) (Hashemi et al., 2025), extending it to support simulation of both normal and

degraded operating conditions. The simulator provides extensive control over operational parameters and component characteristics, enabling the generation of diverse time-series data under varying conditions.

The geothermal system is modeled as multivariate time-series data consisting of 32 operational variables categorized into control inputs and component states. The primary control variables include the producer wellhead pressure (P_{wh}^{prod}), secondary flow rate (q^s), secondary heat exchanger inlet temperature (T_{in}^s), and ESP frequency (f^{esp}).

The remaining variables monitor the thermodynamic and hydraulic state of the process, recording inlet/outlet pressures and temperatures across the heat exchanger, ESP, separators, and filtration units ($Filter_{A/B}$), as well as power metrics (E_{esp}^{el}, $E_{\text{booster pump}}^{el}$ and $Thermal_{PowerGen}$).

All pressures are measured in bar, temperatures in degrees Celsius, flow rates in m^3/h, and frequencies in Hz.

Modeling Assumptions

- No temperature loss prior to the heat exchanger (thermal resistance set to zero).
- No pressure losses in surface pipelines.
- Producer and injector reservoir productivity indices, representing the reservoir's capacity to produce or inject fluid, are set to 20 and 3.4 $\text{m}^3/(\text{bar} \cdot \text{h})$ respectively.
- Producer reservoir pressure and injector reservoir pressure are both set to 265 bar.
- Ambient temperature fixed at 20°C.

The heat exchanger is modeled using a counter-flow configuration, selected for its superior thermal efficiency compared to parallel-flow designs (Chand et al., 2021).

Manual specification of control actions over long temporal horizons is challenging when aiming to capture the variability observed in real plant operations. Defining realistic time-dependence across multiple interlinked variables would require significant expert effort and may still fail to reflect the unpredictable nature of operator behavior. Instead, we employ random control variation to emulate operator driven adjustments during normal plant operation. Although unpredictable, the control ranges and frequencies were consulted with subject matter experts and validated against real world operational data.

3.4 Synthetic Data Generation

Variability is modeled using a Bernoulli-Gaussian process. This method introduces occasional interventions to the system and allows evaluating forecasting models' performance under realistic stress and volatility.

Every minute, each variable has a fixed probability of change. When triggered, the adjustment is sampled from the corresponding distribution and clipped to remain within safe operational bounds:

- **ESP Frequency (f^{esp}):** 0.002% chance per minute of increase or decrease by a value drawn from a uniform distribution $\mathcal{U}(3, 5)$ Hz, clipped to 55–70 Hz. This models very rare operator interventions within typical ESP operating limits. When triggered, the current frequency is adjusted by 0.05% every hour until it reaches the desired value.
- **Flow Rate (q_s):** 0.03% chance per minute of increase or decrease by a value drawn from a Gaussian distribution $\mathcal{N}(0, 0.1)$ m^3/h, clipped to 145–165 m^3/h. This reflects occasional tuning actions with small but realistic magnitude.
- **Inlet Temperature (T_{in}^s):** 0.2% chance per minute of increase or decrease by a value drawn from $\mathcal{N}(0, 0.045)$°C, clipped to 20–30°C. This captures infrequent thermal adjustments while enforcing realistic bounds.
- **Wellhead Production Pressure ($p_{\mathbf{wh}}^{prod}$):** 0.01% chance per minute of increase or decrease by a value drawn from $\mathcal{N}(0, 0.0005)$ bar, clipped to 5–10 bar. This models very rare pressure-related interventions with minimal magnitude.

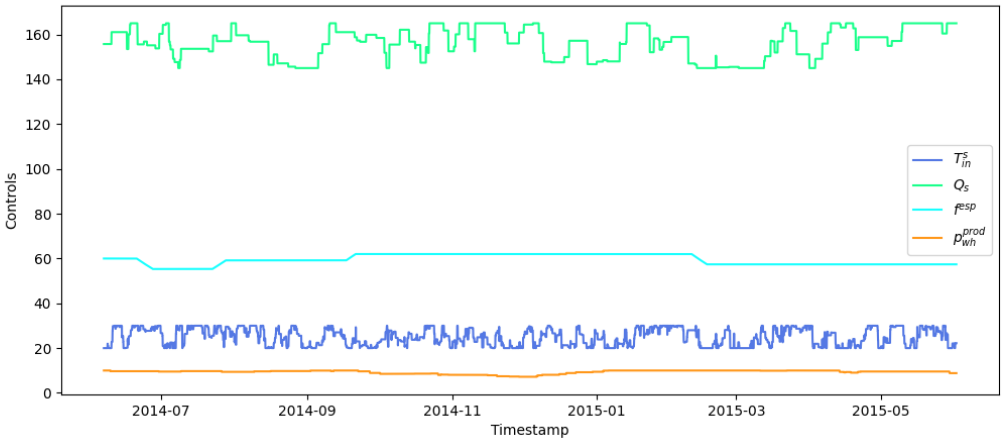


Figure 6: Time series of control variables: illustrating normal operational behavior over 360 days sampled at hourly resolution, highlighting rare and bounded variations in pump frequency (f^{esp}), heat exchanger inlet secondary flowrate (q_s) and temperature (T^s_{in}), and producer wellhead pressure (p^{prod}_{wh}).

The variation is showcased in Figure 6, which visualizes control variables in Dataset 1 (see Section 3.6 for details) over a period of 360 days of normal operation. These small but meaningful adjustments provide a realistic simulation for human intervention without requiring an overly complex model.

3.5 Degradation Modeling

Three degradation events are considered: Electrical Submersible Pump (ESP) degradation, heat exchanger fouling, and filter clogging. Degradation events are evaluated once per day. Each event is represented as beginning from normal operation, progressing through gradual degradation, and terminating in a maintenance action that restores the component to back to normal conditions.

Each component is associated with an explicit degradation rate, which may vary between events. Within a single event, the degradation rate remains constant. Degradation is modeled by gradually increasing a component's resistance parameter, which reduces operational efficiency.

3.5.1 ESP Degradation

The ESP degradation rate is sampled from a uniform distribution $\mathcal{U}(0.5, 0.75)$, resulting in one degradation event per year. When triggered, the ESP resistance is updated as

$$r \leftarrow r \cdot (1 + \mathcal{U}(0.02, 0.05)). \tag{10}$$

This process continues until the ESP resistance increases to 20%, after which the ESP is shut down by setting its operating frequency to zero, and maintenance is initiated. During maintenance, the resistance is reduced hourly using a Bernoulli–Uniform process with probability 0.75 and magnitude $\mathcal{U}(0.05, 0.09)$, until the resistance returns to its normal value. A cooldown period prevents new degradation events from occurring for a fixed number of days following maintenance. Figure 7 visualizes this process from Dataset 3 (see Section 3.6 for details).

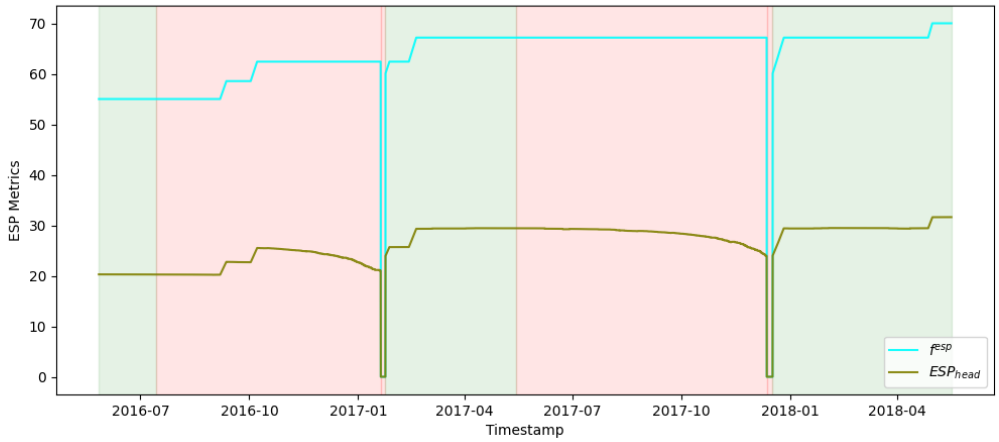


Figure 7: ESP operating frequency and generated head over 720 days, showing two degradation events. Normal operation is indicated in the green region, while degrading operation in the red region. This illustrates degradation manifesting as a gradual decline in pump performance (ESP_{head}), reflecting reduced head generation despite nominal operating frequency (f^{esp}).

3.5.2 Heat Exchanger Fouling

Heat exchanger fouling rates are drawn from $\mathcal{U}(0.5, 0.9)$, producing two fouling events per year. When triggered, the hydraulic resistance increases according to

$$r \leftarrow r \cdot (1 + \mathcal{U}(0.01, 0.03)). \tag{11}$$

This increase leads to higher pressure drops and reduced thermal efficiency, which is represented using an exponential model in which the heat transfer coefficient decays smoothly according to

$$U_{\text{degraded}} \leftarrow U_{\text{clean}} \exp[-k(r - r_0)] \tag{12}$$

where $U_{\text{clean}} = 250e3\ W/K$, $r_0 = 0.001$ (representing clean state resistance), and $k = 120$, representing the decay factor, was calibrated to represent a realistic loss in outlet temperature over the fouling range. Fouling progresses until the pressure drop across the heat exchanger reaches 2 bar, at which point maintenance is triggered. During maintenance, resistance is reduced hourly using a Bernoulli–Uniform process with probability 0.5 ± 0.2 and magnitude $\mathcal{U}(0.01, 0.05)$.

Additionally, these events incorporate severity escalation: after a predefined number of days, all degradation increments are increased by 50%. As with the ESP, a cooldown period is enforced after maintenance before new fouling events may occur.

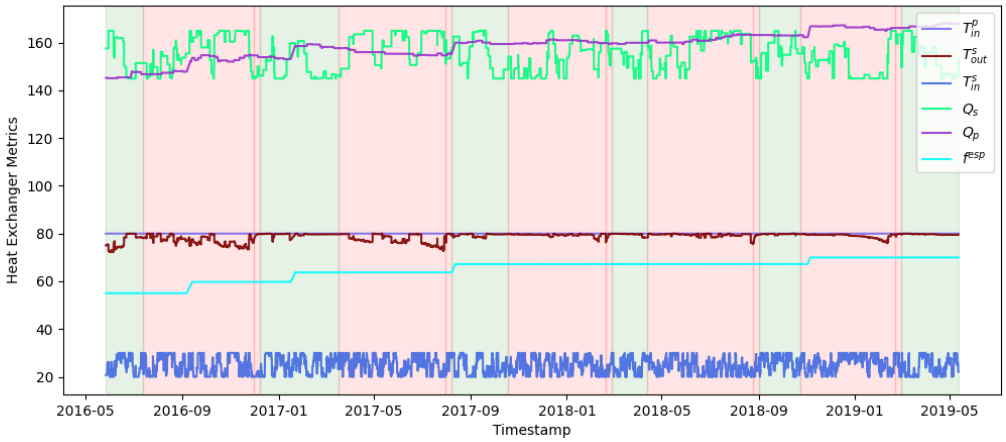


Figure 8: Time series of heat exchanger operating variables during normal and degrading operation. The primary inlet temperature (T^p_{in}) is held constant at 80°C, while variations in primary (Q_p) and secondary (Q_s) flow rates and pump frequency (f^{esp}) interact with fouling-induced resistance. Degraded operation is highlighted in red and normal operation in green.

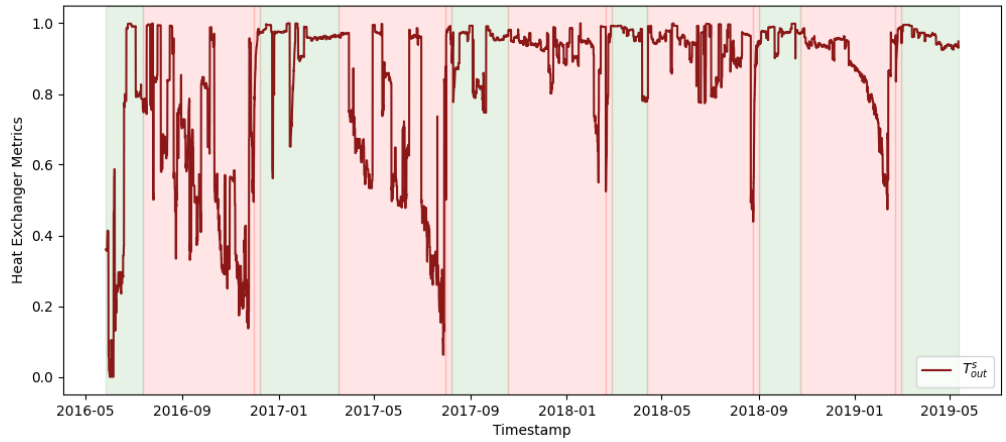


Figure 9: Normalized secondary outlet temperatures (T^s_{in}) over the same period, illustrating the reduction in performance during fouling events (red) relative to normal operation (green). The temperature drop reflects the combined effect of reduced heat-transfer efficiency and the flowrates on the heat exchanger.

Figures 8 and 9 illustrate the impact of heat exchanger fouling on thermal behavior. While the primary inlet temperature (T^p_{in}) is maintained at a constant 80°C, progressive fouling leads to a noticeable decline in secondary outlet temperature (T^p_{in}) , which is also heavily influenced by the flowrates, indicating reduced effectiveness.

3.5.3 Filter Clogging

Two filters are modeled, with Filter A degrading faster than Filter B, since the brine encounters it first. The degradation rates are sampled from $\mathcal{U}(0.8 \pm 0.1)$ and $\mathcal{U}(0.6 \pm 0.1)$, respectively. When triggered, their resistances are increased by $\mathcal{U}(0.1, 0.3)$ and $\mathcal{U}(0.09, 0.2)$, respectively.

Clogging continues until the pressure drop across a filter reaches 0.2 bar, at which point maintenance is initiated. During maintenance, resistances are reduced hourly using a probabilistic process with probability 0.6 ± 0.2 and magnitude $\mathcal{U}(0.01, 0.02)$. As with other components, a cooldown period prevents immediate reoccurrence of degradation. Figure 10 visualizes this process.

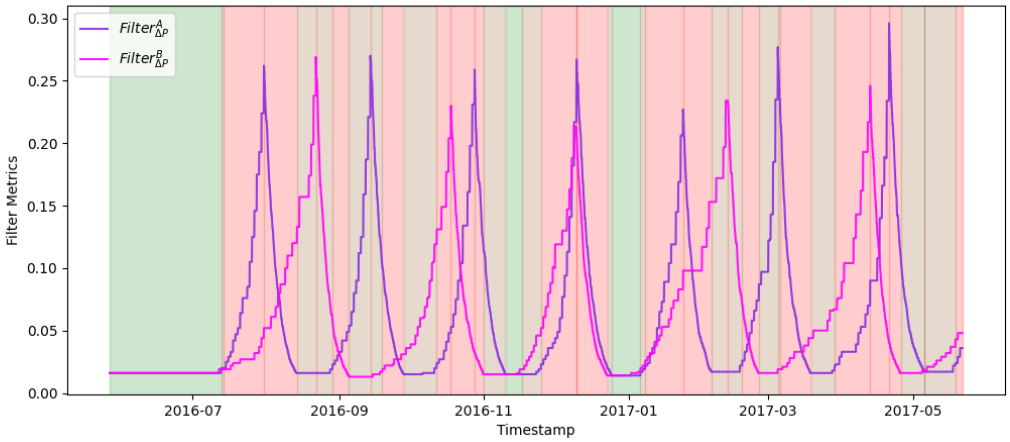


Figure 10: Time series of Filter$_A$ and Filter$_B$ pressure drops (ΔP) across normal (green) and degradation (red) operating conditions.

Overall, this framework models realistic degradation and maintenance patterns, accounting for variations in damage severity, downtime, and repair effectiveness across different components.

3.6 Dataset Information and Visualization

Six datasets are provided at an hourly resolution, obtained by aggregating minute-level data, as shown in Table 3:

Dataset	Condition	Duration	Number of Events	Number of Samples
Dataset 1	Clean	360 days	0	8640
Dataset 2	Clean	360 days	0	8640
Dataset 3	ESP	720 days	2	17280
Dataset 4	HEX	1080 days	5	25920
Dataset 5	Filter	360 days	12	8640
Dataset 6	Combined	1080 days	49	25920

Table 3: Overview of datasets used in the case study benchmark. The first table reports dataset metadata, while the second summarizes dataset conditions and label structure. All datasets contain 32 variables (1 timestamp, 27 operational features, 4 label columns).

Dataset 2 is the continuation of **Dataset 1**, with the exception of a manual adjustment to the ESP frequency (f^{esp}), which is set to 67 Hz at the start of Dataset 2 (Dataset 1 restricts the frequency to a maximum of 62 Hz). Datasets 3-6 are continuations of Dataset 2. By *continuation*, it is implied that both timestamps and the control parameter states are carried forward without reset. All datasets were generated with the same seed (84) to ensure consistency. The datasets will be made publicly available through Nieuwe Warmte Nu (n.d.), and the source code used in this study will be made available upon request.

4 Results and Evaluation

4.1 Forecasting Task

To evaluate generalization under operational regime shift, baseline models are trained on Dataset 1 and tested on Dataset 2. Conversely, foundation models are evaluated directly on Dataset 2 under zero-shot conditions. The target variable is the heat exchanger's secondary-side outlet temperature, representing the delivery fluid temperature for downstream direct-use heating. Using a fixed 7-day context window, performance is measured across 24, 48, and 96-hour horizons to verify robustness.

Table 4 reports mean performance across forecasting horizons. Standard deviations are computed over 5 independent runs, capturing variability arising from model initialization, and are reported in

	Horizon: 24		Horizon: 48		Horizon: 96	
Model	**R^2**	**RMSE**	**R^2**	**RMSE**	**R^2**	**RMSE**
Naive	0.894	0.681	0.789	0.957	0.588	1.338
RandomForest	0.805	0.923	-	-	-	-
LSTM	0.923	0.581	0.872	0.742	0.791	0.947
TiDE	**0.971**	2.942	**0.986**	3.024	0.824	2.879
Chronos	0.967	**0.378**	0.944	**0.493**	**0.875**	**0.737**
Moirai	0.879	0.727	0.760	1.021	0.522	1.442
TimesFM*	0.879	0.727	0.777	0.984	0.547	1.404
Chronos*	0.860	0.780	0.766	1.009	0.453	1.542
Moirai*	0.880	0.722	0.771	0.998	0.539	1.414

Table 4: Forecasting performance on heat exchanger secondary outlet temperature (Dataset 2) across horizons (24, 48, 96). Results report covariate-aware forecasting unless otherwise indicated. Models marked with * denote univariate forecasting. Best values per column are in bold.

Table 8, and inference runtimes, measured once per model, are reported in Table 9 (both in Appendix A). Chronos (covariate-aware) achieves the lowest RMSE across all horizons (0.378, 0.493, 0.737 at 24, 48, and 96 hours) and the highest R^2 at the longest horizon (0.875), making it the most accurate model in this evaluation. LSTM ranks second in RMSE across horizons and shows gradual degradation with increasing horizon, reaching $R^2 = 0.791$ and RMSE $= 0.947$ at 96 hours. The Naive baseline performs well at short horizons ($R^2 = 0.894$ at 24 hours) but degrades at longer horizons ($R^2 = 0.588$ at 96 hours), reflecting increasing forecasting difficulty.

TiDE achieves the highest R^2 at horizons 24 and 48 (0.971 and 0.986), while producing substantially higher RMSE values across all horizons (2.942, 3.024, and 2.879), exceeding all other models. Among zero-shot foundation models, the effect of covariates varies across architectures. Chronos shows a clear performance drop in univariate mode, with R^2 decreasing from 0.967 to 0.860 at horizon 24 and to 0.453 at horizon 96, indicating strong dependence on exogenous inputs. Moirai shows only minor differences between multivariate and univariate configurations across all horizons, suggesting limited use of covariate information.

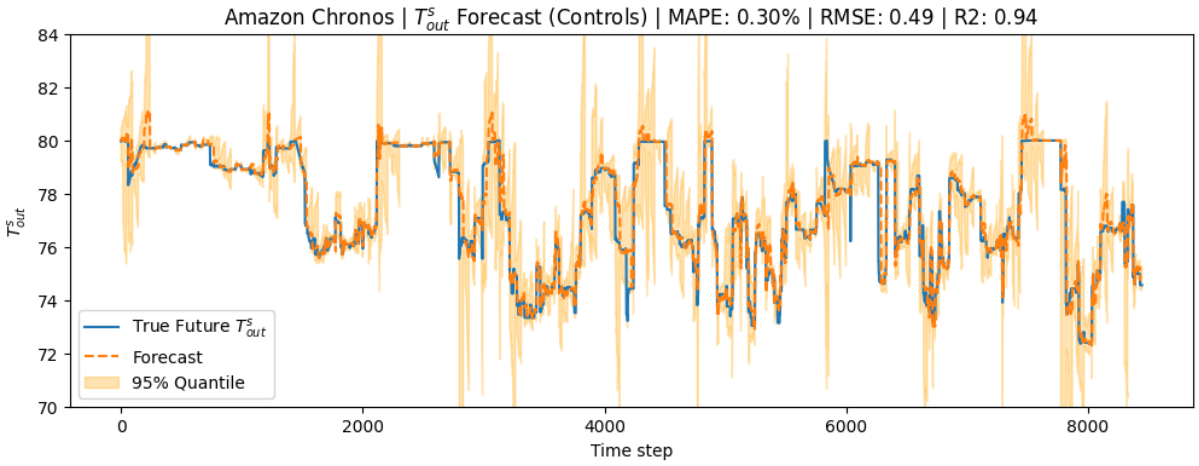


Figure 11: Chronos forecasting at horizon 48, showcasing strong forecasting accuracy and low prediction error, explaining it's low RMSE across all horizons

TiDE's high R^2 at shorter horizons suggests strong capture of variance in the target signal; however, its high RMSE reflects the MLP-based architecture's emphasis on global trend fitting rather than pointwise accuracy, leading to large deviations at individual timesteps. Figure 12 confirms this behavior: TiDE follows the overall signal trend but shows occasional large errors at specific time points. Chronos produces more stable forecasts, as shown in Figure 11, which explains its lower RMSE despite comparable R^2 values. The strong performance of LSTM under regime shift suggests that learned temporal dependencies generalize reasonably well across operating conditions, although the performance gap relative to Chronos

indicates remaining limitations.

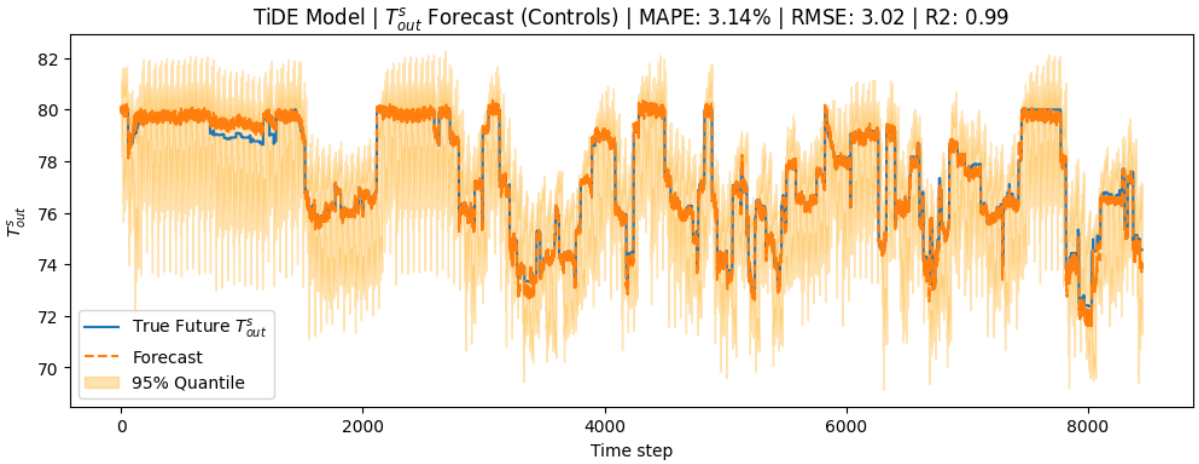


Figure 12: TiDE forecasting at horizon 48, showcasing strong variance capture but weaker pointwise accuracy, resulting in high R^2 yet significantly high RMSE driven by its architecture's emphasis on global trend capture rather than pointwise alignment.

The under-performance of univariate zero-shot foundation models relative to the Naive baseline highlights a key limitation: without covariates, such models cannot distinguish operating regimes from the temperature signal alone. The stark contrast between Chronos' multivariate and univariate performance (Appendix B) confirms that covariates provide critical predictive information missing from the target series. Conversely, Moirai shows negligible differences between configurations (Appendix B), indicating poor covariate utilization. Ultimately, effectively leveraging exogenous inputs is vital for long-horizon forecasting in this system.

Overall, these results show that pretrained foundation models, in particular Chronos, achieve strong forecasting performance when covariates are integrated effectively, outperforming trained baselines such as LSTM and TiDE. This is notable because Chronos operates in a zero-shot setting without task-specific fine-tuning, indicating that its architecture generalizes well to this industrial forecasting task.

4.2 Anomaly Detection Tasks

This section evaluates anomaly detection performance on the three degradation scenarios discussed in Section 3. Each scenario targets a different component and is used to assess model robustness across distinct failure modes. The following subsections introduce the specific evaluation setups.

4.2.1 Heat Exchanger Fouling

Baseline models are trained on Dataset 1, as in the forecasting task. Evaluation is performed on Dataset 2 and Dataset 4 using a sliding-window inference procedure. A context window of 7 days (168 hourly observations) is used, consistent with the forecasting setup, and only a 24-hour horizon is considered.

The anomaly detection target is derived from raw temperature measurements through the Number of Transfer Units (NTU), defined as

$$\varepsilon_t = \frac{q_s \left(T^s_{out} - T^s_{in}\right)}{\min\left(q_p, q_s\right)\left(T^p_{in} - T^s_{in}\right)}, \qquad \text{NTU} = -\ln(1 - \varepsilon + 10^{-7}). \tag{13}$$

NTU combines both inlet and outlet temperatures into a single value that directly represents heat exchanger effectiveness, making it a more interpretable indicator of fouling than outlet temperature alone, and is the standard industry metric for this purpose (Andrijić et al., 2021).

For Random Forest and MLP classifiers, the training and evaluation split is performed on the concatenation of Dataset 2 and Dataset 4 in temporal order. The resulting sequence is split chronologically, such that approximately 54.5% of the combined data (corresponding to all of Dataset 2 and the initial

segment of Dataset 4) is used for training, while the remaining 45.5% is used for evaluation. The unsupervised methods, on the other hand, are trained and calibrated exclusively on healthy data (Dataset 2) and evaluated directly on Dataset 4.

	Forecasting		**Novelty Detection**			**Classifier (RF)**		
Model	**R^2**	**RMSE**	**P**	**R**	**F1**	**P**	**R**	**F1**
RandomForest	-0.862	1.602	**0.720**	**0.864**	**0.786**	0.251	0.550	0.345
LSTM	0.771	0.561	0.565	0.644	0.602	0.847	0.701	0.767
TiDE	**0.885**	1.546	0.661	0.613	0.636	0.731	0.701	0.716
Chronos	0.845	0.461	0.686	0.604	0.642	0.875	0.705	0.781
Moirai	0.862	**0.435**	0.631	0.607	0.619	**0.928**	0.702	**0.800**
TimesFM	0.848	0.456	0.594	0.674	0.632	0.823	**0.722**	0.776
Embedding-Based Anomaly Detection								
Model			**Isolation Forest**			**Classifier (MLP)**		
LSTM			0.635	**0.714**	**0.672**	**0.794**	**0.723**	**0.757**
TiDE			0.665	0.603	0.632	0.592	0.664	0.626
Chronos			0.542	0.245	0.337	0.638	0.616	0.627
Moirai			0.564	0.603	0.583	0.697	0.632	0.663
TimesFM			**0.751**	0.604	0.670	0.759	0.627	0.686

Table 5: Forecasting and anomaly detection performance on the heat exchanger task. The top section reports forecasting performance (R^2, RMSE) and forecasting-based anomaly detection results. The bottom section reports embedding-based anomaly detection performance. Anomaly detection performance is evaluated using Precision (P), Recall (R), and F1-score (F1). Higher values indicate better performance for all metrics except RMSE, for which lower values are better. Best values per column are shown in bold.

Table 5 summarizes the mean performance on the heat exchanger forecasting and anomaly detection tasks. Standard deviations, computed over 5 independent runs, are reported in Appendix C.1, together with inference runtimes and hyperparameter settings for embedding-based methods. In forecasting, foundation models lead: Moirai achieves the lowest RMSE of 0.435, closely followed by TimesFM at 0.456 and Chronos at 0.461. LSTM remains competitive with an RMSE of 0.561, while TiDE yields a high R^2 of 0.885 but poor RMSE of 1.546. Random Forest forecasts poorly with an R^2 of -0.862 and RMSE of 1.602. However, for forecasting-based novelty detection, Random Forest achieves the highest F1-score of 0.786, while other models cluster between 0.60 and 0.64. In the forecasting-based classifier approach, Random Forest drops to a low 0.345 F1-score, with other models performing comparably. For embedding-based Isolation Forest detection, Chronos yields the lowest F1-score of 0.337 against a 0.58–0.67 range for the other models. Conversely, the MLP-based classifier delivers the most consistent results across all models, spanning F1-scores of 0.62–0.75.

TiDE's RMSE is consistent with its behavior observed in prior tasks: the model captures the overall trend of the NTU signal but struggles with accurate pointwise predictions, explaining the divergence between its strong R^2 and poor RMSE. Random Forest fails to produce meaningful forecasts, and LSTM, while competitive, remains slightly behind the foundation models. Together, these results reinforce the forecasting advantage of pretrained foundation models over task-specific baselines.

In forecasting-based novelty detection, this ranking inverts: Random Forest achieves the highest F1-score of 0.786 despite its weak forecasting. This stems from the fault's gradual degradation nature. Because Random Forest cannot adapt to slow distributional shifts, the actual NTU observations consistently fall outside its prediction intervals. This triggers continuous anomaly flags, yielding high recall through indiscriminate alarming rather than true degradation tracking. Conversely, superior forecasting models adapt to gradual shifts, treating the degradation as the new normal and suppressing anomaly flags. This is illustrated in Figure 13, where foundation models maintain low, stable errors throughout the degradation period, while Random Forest exhibits volatile error margins.

The forecasting-based Random Forest classifier reverses this trend, as performance scales directly with forecast quality. Random Forest's poor forecasts translate into a sharp F1 drop to 0.345: a reduction of 40% from the novelty setting. Foundation models and LSTM yield stronger results with F1-scores of

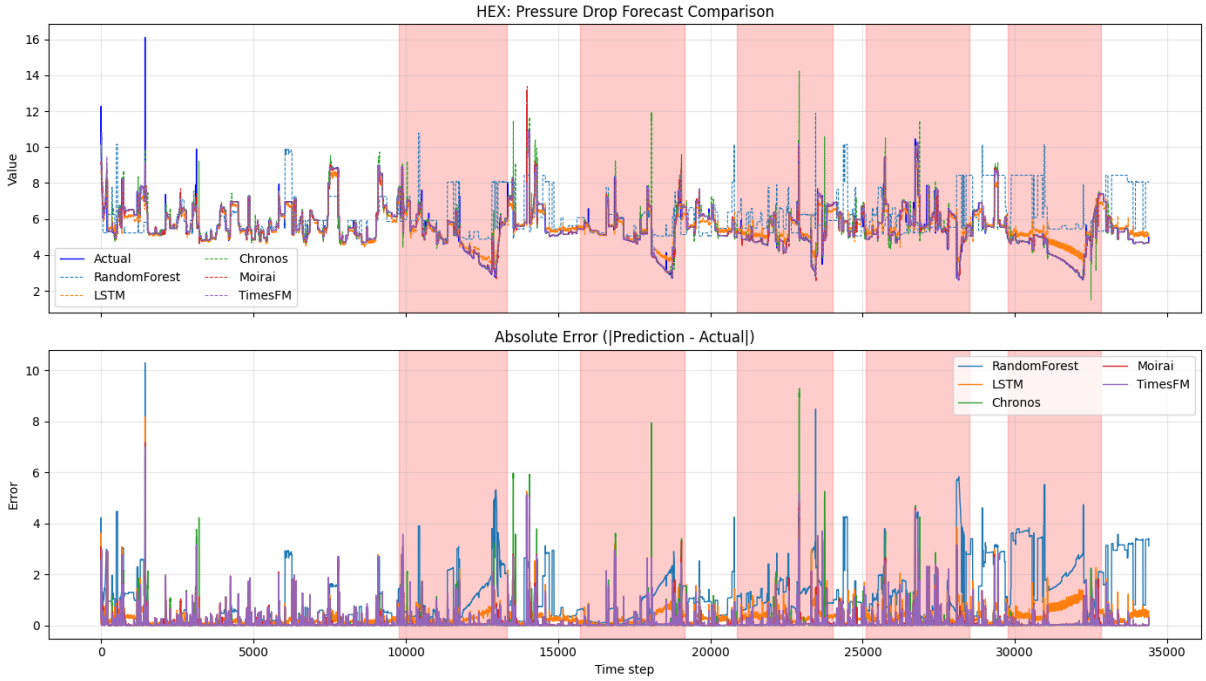


Figure 13: A dual-plot showing actual vs. predicted values (top) and absolute errors (bottom). Foundation models show superior pointwise stability compared to the more volatile error margins of the Random Forest and LSTM baseline. The red regions indicate anomalous regions.

0.716-0.800, led by Moirai at 0.800. However, generalization remains limited; as classifier accuracy on held-out evaluation data ranges from 0.367 (Random Forest) to 0.685 (TimesFM), compared to 0.522 using ground-truth NTU observations. This gap confirms classifier sensitivity to both forecast quality and training data representativeness.

For embedding-based Isolation Forest detection, most models yield consistent results with F1-scores in range of 0.58-0.67, except Chronos, which drops to 0.337 due to low recall of 0.245. This is likely caused by the high dimensionality (4608) and cross-attention structure of Chronos embeddings, which compress into a less separable latent space for the unsupervised method. The MLP classifier largely recovers this gap, achieving F1-scores of 0.62-0.76, indicating that fault-relevant features are present but require supervised boundaries to extract. LSTM leads the MLP approach with its F1-score of 0.757, potentially due to the easier separability of its lower dimensional embeddings (128), though generalization remains constrained with test accuracies spanning 0.50 (Chronos) to 0.713 (TiDE).

Together, these results reveal two main limitations. First, unsupervised novelty detection becomes less sensitive during gradual degradation, as forecasting models adapt to slowly changing system behavior. Second, supervised classifiers avoid this issue but require sufficiently labeled datasets, which are rarely available in industrial settings.

4.2.2 Electrical Submersible Pump Degradation

Baseline models are trained on Dataset 1, as in the forecasting task. Evaluation is performed on Dataset 2 and Dataset 3 using the same sliding-window inference setup as in the heat exchanger task. For classifier-based methods, a 67.5%/32.5% train/evaluation split is used on labeled fault data. Unsupervised methods follow the same setup as in the previous task. The anomaly detection target is ESP_{head}, representing pump efficiency under varying operating conditions.

Table 6 summarizes the mean performance on the electrical submersible pump task. Standard deviations, computed over 5 independent runs, inference run-times and hyperparameter settings for embedding-based methods are provided in Appendix C.2. In forecasting, foundation models dominate: Chronos performs best with an R^2 of 0.918 and an RMSE of 1.233, followed by Moirai with an R^2 of 0.890 and an RMSE of 1.431, and TimesFM with an R^2 of 0.849 and an RMSE of 1.676. LSTM is a competitive baseline with an RMSE of 2.162, whereas TiDE yields a high R^2 of 0.878 but a poor RMSE of 6.107. Random Forest performs worst in forecasting with an R^2 of 0.339 and an RMSE of 3.510. For forecasting-based

	Forecasting		Novelty Detection			Classifier (RF)		
Model	**R^2**	**RMSE**	**P**	**R**	**F1**	**P**	**R**	**F1**
RandomForest	0.339	3.510	0.324	**1.000**	0.489	**0.677**	**0.633**	**0.654**
LSTM	0.749	2.162	0.552	0.672	0.606	0.203	0.600	0.304
TiDE	0.878	6.107	**0.894**	0.811	**0.851**	0.048	0.600	0.089
Chronos	**0.918**	**1.233**	0.772	0.604	0.677	0.365	0.600	0.454
Moirai	0.890	1.431	0.603	0.607	0.605	0.331	0.600	0.427
TimesFM	0.849	1.676	0.559	0.406	0.471	0.324	0.600	0.421
Embedding-Based Anomaly Detection								
Model			**Isolation Forest**			**Classifier (MLP)**		
LSTM			**0.817**	**0.821**	**0.819**	0.199	0.609	0.300
TiDE			0.778	0.608	0.683	0.666	0.602	0.632
Chronos			0.272	0.312	0.291	0.600	**0.637**	0.618
Moirai			0.357	0.606	0.449	0.700	0.621	0.658
TimesFM			0.324	0.600	0.421	**0.705**	0.629	**0.665**

Table 6: Forecasting and anomaly detection performance on the electrical submersible pump task. The top section reports forecasting metrics and forecasting-based anomaly detection results. The bottom section reports embedding-based anomaly detection performance. Best values per column are in bold.

novelty detection, TiDE leads with an F1-score of 0.851, outperforming Chronos with 0.677, LSTM with 0.606, and Moirai with 0.605, while Random Forest with 0.489 and TimesFM with 0.471 lag behind. In the forecasting-based classifier setting, Random Forest performs best with an F1-score of 0.654, while LSTM drops to 0.304 and TiDE drops further to 0.089. In embedding-based anomaly detection via Isolation Forest, LSTM achieves the highest F1-score of 0.819, while Chronos performs worst with 0.291. Conversely, the embedding-based MLP classifier yields the most consistent performance, with F1-scores ranging from 0.618 for Chronos to 0.665 for TimesFM.

TiDE's high R^2 yet poor RMSE is consistent with earlier observations: the model captures the overall trends but fails to model individual predictions. Random Forest underperforms globally, and LSTM serves as a solid baseline that still trails behind the foundation models. These results again reinforce the forecasting advantage of pretrained foundation models in capturing complex temporal relationships without any task-specific fine-tuning.

In forecasting-based novelty detection, the performance ranking partially inverts. TiDE achieves the highest F1-score of 0.851 despite its weak pointwise forecasting accuracy; its high RMSE 6.107 stems from persistent deviations during regime shifts, which inherently generates more anomaly flags. Similarly, the LSTM baseline matches foundation model anomaly detection performance, likely by failing to track the degradation trend. While this lack of adaptation penalizes its forecasting metrics, the resulting sustained deviation from actual values provides the consistent flags necessary for detection. Conversely, the stronger forecasters (Chronos, Moirai, TimesFM) adapt smoothly to gradual degradation, reducing discrimination between normal and anomalous operating regimes. Figure 14 illustrates this trade-off: superior forecasting accuracy actively diminishes anomaly detection sensitivity under slow, progressive faults.

The forecasting-based classifier performs poorly overall, with the sole exception of Random Forest, achieving F1-score of 0.654. This failure stems from the classifier's poor generalizability, as evidenced by low test accuracies across all inputs: 0.426 on actual data, 0.448 on RF, 0.402 on LSTM, 0.40 on TiDE, 0.42 on Chronos, 0.42 on Moirai, and 0.41 on TimesFM. A contributing factor is the limited fault events for this case available during training: the classifier is trained only train on a single fault event and is evaluated on a different unseen event. This may be constraining its ability to learn decision boundaries and limiting its generalizability. This performance decline highlights a key limitation of standard classifiers trained on forecast outputs. The classifier fails to separate normal and anomalous operating regimes; a shift that appears obvious to a human observer is likely misinterpreted by the model as a routine operational change. Additionally, this bottleneck reinforces the dependency on labeled data quality noted previously.

For embedding-based Isolation Forest detection, LSTM achieves the highest F1-score of 0.819, while

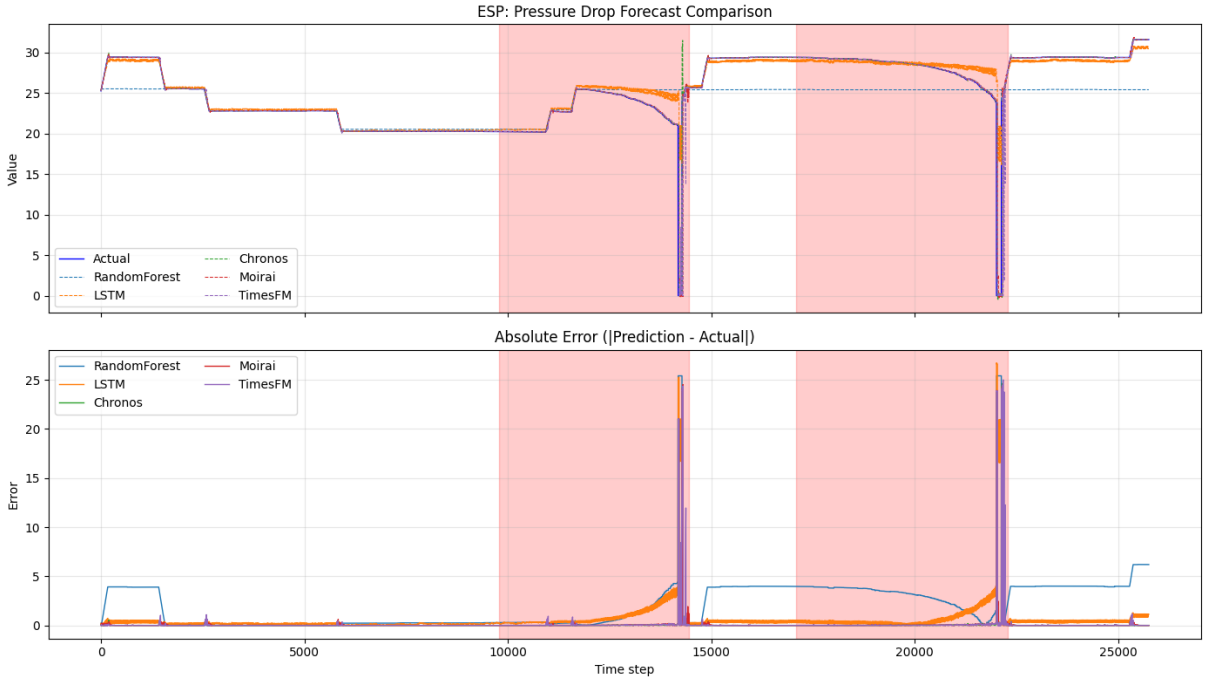


Figure 14: ESP Head forecasting comparison across models at a 24-step horizon, showcasing that models capturing overall signal trends can still enable effective anomaly detection despite comparatively weak forecasting performance, while more accurate forecasts do not always yield the highest detection performance.

Chronos performs worst with 0.291. LSTM produces relatively compact, lower-dimensional representations that preserve local temporal variation, making anomalies more separable under unsupervised partitioning. In contrast, Chronos embeddings are higher-dimensional, which can blur local separability in the latent space and reduce Isolation Forest effectiveness. The remaining foundation models fall between these extremes, reflecting a trade-off between global representation expressiveness and local anomaly separability.

The embedding-based MLP classifier yields stable performance across foundation models, with F1-scores tightly spanning 0.618–0.665 and classifier accuracies reaching 0.83–0.90 (Chronos, Moirai, TimesFM). This indicates again that discriminative information is consistently present in the latent space, but requires supervised decision boundaries to be extracted. Notably, Chronos achieves competitive results here despite its weak Isolation Forest performance, confirming again that its embeddings are highly informative but poorly suited for unsupervised separation. Conversely, LSTM showcases the weakest F1-score of 0.300 and classifier accuracy of 0.39, while TiDE achieves just 0.48 classifier accuracy. This sharp divide suggests that models with weaker forecasting performance may produce embeddings which are less representative of the system state, reducing reliability under supervised learning.

The ESP fault results reinforce findings from earlier tasks while adding additional insights. Forecasting-based novelty detection remains sensitive to the forecasting–accuracy trade-off, where stronger forecasters adapt to gradual degradation and reduce anomaly sensitivity. Similarly, forecasting-based classifiers fail across most models, indicating limited generalization when labeled fault examples are scarce and decision boundaries are affected by gradual regime shifts. Embedding-based Isolation Forest shows inconsistent performance largely influenced by the structure of the latent space rather than representation quality. The embedding-based MLP classifier is the most robust approach, consistently separating normal and anomalous conditions across all foundation models and achieving accuracies up to 0.90 despite limited fault data. This supports the finding that TSFM embeddings encode meaningful system information that supervised methods can effectively exploit, even when other detection strategies fail.

4.2.3 Filter Clogging

Baseline models are trained on Dataset 1, as in the forecasting task. Evaluation is performed on Dataset 2 through Dataset 5 using the same sliding-window inference setup as in the previous tasks. For classifier-

based methods, a 78.3%/21.7% train/evaluation split is used on labeled fault data. Unsupervised methods follow the same setup as in the previous tasks. The anomaly detection target is $Filter^{A}_{\Delta P}$, representing the pressure drop across the filter and serving as an indicator of filter condition.

	Forecasting		**Novelty Detection**			**Classifier (RF)**		
Model	**R^2**	**RMSE**	**P**	**R**	**F1**	**P**	**R**	**F1**
RandomForest	-0.189	0.051	0.075	**1.000**	0.141	0.000	0.000	0.000
LSTM	-0.149	0.051	0.272	0.993	0.427	0.939	0.692	0.797
TiDE	0.955	0.066	0.487	0.941	0.642	0.611	0.843	0.708
Chronos	**0.986**	**0.005**	**0.551**	0.941	**0.695**	**1.000**	**0.850**	**0.919**
Moirai	0.970	0.008	0.534	0.971	0.689	0.785	0.816	0.800
TimesFM	0.976	0.007	0.321	0.980	0.484	**1.000**	0.692	0.818
Embedding-Based Anomaly Detection								
Model			**Isolation Forest**			**Classifier (MLP)**		
LSTM			0.015	**1.000**	0.031	**1.000**	**0.993**	**0.996**
TiDE			0.081	**1.000**	0.150	0.986	0.986	0.986
Chronos			0.800	0.679	0.734	0.981	0.856	0.870
Moirai			**0.946**	0.938	0.942	0.984	0.909	0.945
TimesFM			0.926	**1.000**	**0.961**	0.818	0.872	0.844

Table 7: Forecasting and anomaly detection performance on the filter task. The top section reports forecasting metrics and forecasting-based anomaly detection results. The bottom section reports embedding-based anomaly detection performance. Best values per column are in bold.

Table 7 summarizes mean performance on the filter task. Standard deviations, computed over 5 independent runs, inference run-times and hyperparameter settings for embedding-based methods are provided in Appendix C.3. In forecasting, foundation models lead: Chronos performs best with R^2 of 0.986 and RMSE 0.005, followed by TimesFM with R^2 of 0.976 and RMSE of 0.007 and Moirai with R^2 of 0.970 and RMSE of 0.008. TiDE is strong with R^2 of 0.955 but has a higher RMSE of 0.066, while LSTM and Random Forest perform poorly with negative R^2 scores and RMSEs around 0.051. For forecasting-based novelty detection, Chronos leads with an F1-score of 0.695, closely followed by Moirai at 0.689 and TiDE at 0.642, while TimesFM at 0.484, LSTM at 0.427 and Random Forest at 0.141 lag. In the forecasting-based classifier, Chronos again tops performance with a F1-score of 0.919, followed by TimesFM at 0.818, Moirai at 0.800, LSTM at 0.797 and TiDE at 0.708, while Random Forest fails completely with F1-score of 0.000. For embedding-based Isolation Forest detection, performance varies: TimesFM leads with F1-score of 0.961, followed by Moirai at 0.942 and Chronos at 0.734, whereas TiDE achieves 0.150 and LSTM achieves 0.031, despite perfect recall. Conversely, the embedding-based MLP classifier delivers consistently high performance across all models, spanning from 0.844 F1-score for TimesFM, up to 0.996 for LSTM, with TiDE at 0.986 and Moirai at 0.945 also showing strong results.

The complete failure of Random Forest and LSTM (negative R^2 scores) in forecasting indicates that the filter signal contains complex temporal structures that tree-based and standard recurrent approaches cannot capture without sufficient exposure during training. TiDE's strong R^2 of 0.955 alongside a higher RMSE of 0.066 compared to foundation models is again consistent with its pattern across tasks. The uniformly high recall in novelty detection, 0.941–1.000, across all models indicates that the filter fault manifests as a sharp, sudden deviation rather than a gradual drift, making it easily detectable regardless of forecast quality. Precision is therefore the key differentiator: stronger forecasters maintain tighter prediction intervals during normal operation, minimizing false positives. Conversely, Random Forest and LSTM trigger alarms almost indiscriminately, achieving near-perfect recall at the cost of near-zero precision, rendering their detection outputs unusable.

The Random Forest classifier yields a F1-score of 0.000 for the Random Forest model despite its forecasts being no worse than the LSTM. It produces residuals that carry no discriminative information for a supervised classifier, yielding zero true positive classifications. Chronos and TimesFM achieving perfect precision of 1.000 in the classifier confirms that their residuals are highly structured and cleanly separable, which is consistent with their strong forecasting performance on this task. Figure 15 showcases this difference in forecasting.

The Isolation Forest failure for LSTM and TiDE (F1 = 0.031 and 0.150) despite their near-perfect

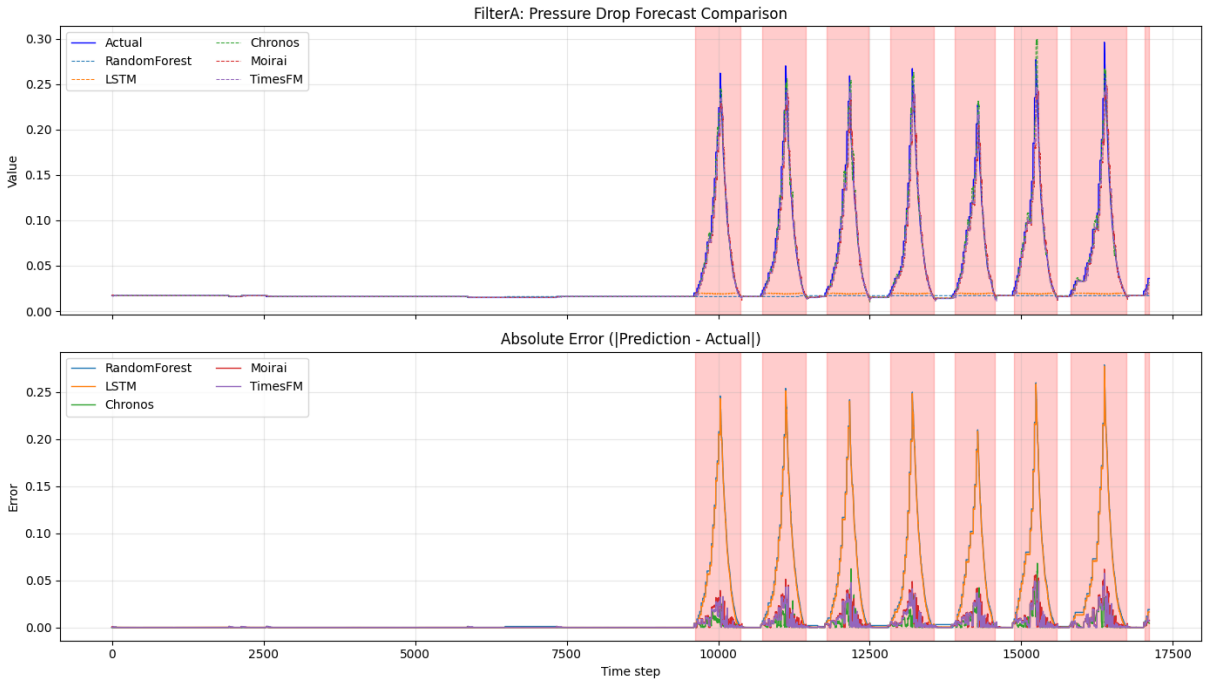


Figure 15: Filter A pressure drop forecasting across models at a 24-step horizon, showcasing that accurate tracking of the rising degradation trend enables near-perfect anomaly detection, while poor forecasts fail to trigger the classifier.

MLP classifier results (F1 = 0.996 and 0.986) is the sharpest observation in this section, mirroring the Chronos pattern from previous two tasks. While the fault information is clearly encoded in their embeddings and linearly extractable via supervised learning, it is too irregular for the unsupervised method to exploit. Conversely, foundation model embeddings, particularly TimesFM and Moirai, are structured linearly enough for Isolation Forest to isolate the fault regime entirely without labels, representing a significant advantage for zero-shot deployment. Finally, the uniformly higher MLP performance on this task compared to previous scenarios suggests the filter fault produces globally distinctive operational signatures that are significantly easier to learn, regardless of the underlying embedding source.

Taken together, these results further reinforce that detection strategy and fault characteristics interact strongly, and that supervised classification on TSFM embeddings remains the most reliable and consistent approach across varying fault regimes.

5 Discussion

5.1 Forecasting

The forecasting results demonstrate a clear advantage of TSFMs over conventionally trained models. Despite requiring no task-specific training data, TSFMs adapt effectively to the volatile control-driven dynamics of geothermal plants, a setting where conventional models, trained on historical plant data, showcase worse performance in the evaluation distribution.

Among the TSFMs, Chronos achieves the strongest performance, significantly outperforming all conventional baselines. This is mainly attributed to how each model handles multivariate relationships. Chronos incorporates covariates directly within the attention mechanism, allowing it to directly leverage control variables as meaningful contextual signals. Moirai encodes all features and the target into a single unified representation before attention is applied, which appears to compress and lose covariate information. TimesFM does not incorporate covariates at all. This leads to both Moirai and TimesFM performing slightly below the naive model baseline. This aligns with their architectural design, as both these models were originally designed for univariate forecasting.

These differences point to a broader insight: while prior work (see Table 1) has often emphasized finetuning as a necessary step to effectively deploy TSFMs on downstream tasks (Mulayim et al., 2025), our results suggest that architecture is more fundamental for effective task adaptation. Chronos was

designed to incorporate covariates, and this single architectural choice is sufficient for it to outperform every competing model without any task-specific supervision.

The importance of multivariate relationships in control-driven systems become increasingly clear as the horizon lengths are increased, a setting where pattern-based models fail entirely, yet Chronos maintains strong performance. Accurate multi-step forecasts carry direct operational value, enabling informed decision-making and allowing operators to anticipate the downstream effects of control changes, positioning TSFMs as a strong candidate for deployment in geothermal condition monitoring.

In contrast, Chronos in the univariate setting, Moirai in both multivariate and univariate setting, and TimesFM in its default univariate setting exhibit performance comparable to or worse than a naive persistence baseline. These results indicate that, depending on the forecast horizon, the models fail to move beyond trivial persistence behavior. Given the substantial computational cost associated with pretraining these large foundation models, such behavior highlights a key practical limitation when multivariate dependencies are not effectively leveraged.

5.2 Anomaly Detection

Anomaly detection results revealed considerably more nuance than the forecasting task. Performance varied widely across both models and tasks, yet several consistent patterns emerged.

Foundation models achieved the strongest forecasting performance across all three tasks, but forecast quality did not consistently translate into anomaly detection performance. This disconnect is driven from a fundamental tension between the two objectives: strong forecasters adapt to gradual degradation, reducing residual errors and suppressing the very signals that anomaly detection relies on. As a result, the best forecaster is not necessarily the best anomaly detector. Novelty detection is especially susceptible to this, as it operates by testing whether actual observations fall within the model's predicted uncertainty bounds. When a model adapts to slow degradation and treats it as the new normal, anomaly flags are suppressed precisely when they are most needed.

This behavior is, however, strongly task-dependent. In the filter clogging task, where degradation was abrupt and clearly distinguishable, foundation models held a clear advantage due to their tight prediction intervals and low false positive rates. In contrast, for the heat exchanger and ESP tasks, where degradation was gradual and slow-moving, trained baselines such as Random Forest and LSTM sometimes outperformed foundation models in novelty detection, not because they tracked the signal better, but because their failure to adapt produced persistent deviations that triggered consistent anomaly flags.

Supervised methods were generally strong when sufficient labeled data was available. The Random Forest classifier achieved competitive results in two of the three tasks, but its performance was heavily dependent on its generalizability. When the classifier successfully learned a separable decision boundary between normal and anomalous regimes, results were strong. However, as observed in the ESP task, poor generalization caused the classifier to fail entirely, producing poor scores regardless of the input source.

Embedding-based methods offered a more reliable alternative to forecasting-based approaches. Across all three tasks, the supervised MLP classifier applied to foundation model embeddings yielded the most consistent performance, reinforcing the conclusion that these embeddings encode rich, fault-relevant information. However, this information is not readily accessible to unsupervised methods: Isolation Forest results were variable and often unreliable, while the MLP classifier consistently extracted meaningful structure when provided with labeled supervision. This points to a key practical constraint: the discriminative content within these embeddings requires explicit labeling to utilize.

Different models also encode information in fundamentally different ways. Chronos embeddings proved highly informative under supervised learning yet performed poorly with Isolation Forest across multiple tasks, suggesting that while the embeddings are rich, their structure is not geometrically separable in a way that unsupervised clustering can exploit. Foundation models such as Moirai and TimesFM, by contrast, produced embeddings that were sufficiently structured for Isolation Forest to achieve strong zero-shot detection in the filter task, representing a meaningful advantage for label-free deployment scenarios.

Taken together, these results highlight two recurring limitations. Unsupervised novelty detection becomes less effective during gradual faults because the forecasting model adapts to the slowly changing behavior, making the abnormal patterns harder to distinguish from normal operation. Supervised classifiers avoid this issue, but they rely on having enough labeled examples of different fault types, which is

often not available in real industrial settings.

5.3 Limitations

There are several limitations that affect the generalizability of this study.

First, all experiments are conducted on data generated using a modified version of the GEMINI Geothermal Digital Twin. While the high-fidelity nature of this twin provides valuable insights into control-driven industrial dynamics, it cannot fully replicate the noise and unpredictability inherent to physical deployments.

Second, all foundation models are evaluated exclusively in a zero-shot setting. This choice is intentional, as it reflects the primary focus of assessing generalization without task-specific adaptation. However, prior work (Marconi, 2025; Park et al., 2025) has shown that fine-tuning can yield additional performance gains, particularly in covariate-rich settings.

Third, the study evaluates three Time Series Foundation Models with different design assumptions. Moirai and TimesFM were primarily developed for univariate forecasting, whereas Chronos is explicitly designed to incorporate covariates. As a result, the comparatively stronger performance of Chronos in this study is therefore not unexpected.

Fourth, anomaly detection is evaluated under a fixed forecasting horizon and fixed context length, whereas forecasting experiments consider multiple horizons. The effect of varying context lengths is not explored, although it may influence sensitivity to gradual degradation patterns.

Fifth, while basic hyperparameter tuning is performed for embedding-based anomaly detection methods, a more systematic analysis may further improve performance across tasks.

Finally, inference run-times are reported for all models and tasks, but the underlying factors contributing to these differences are not explicitly analyzed. Understanding and isolating the reasons for these run-times, such as model architecture, or internal data-flow pipelines, could hold meaningful revelations for further model development.

6 Conclusion and Future Work

This work evaluated pretrained time series foundation models against task-specific baselines across forecasting and anomaly detection tasks in an industrial geothermal setting, spanning heat exchanger fouling, electrical submersible pump degradation, and filter clogging. To support reproducibility and future benchmarking, the datasets generated via the geothermal digital twin are made publicly available, addressing the absence of existing benchmarks that combine meaningful control variables, operational variability, and labeled degradation events in realistic geothermal settings.

In forecasting, Chronos consistently delivered the strongest performance across all tasks and horizons, outperforming both trained baselines and competing foundation models. Its advantage was more significant at longer horizons, where the limitations of task-specific models became apparent. For operational decision-making, this matters directly: longer-horizon forecasting enables earlier intervention, giving operators more time to respond to developing faults before they escalate. This result carries a broader implication: covariate-aware forecasting is not merely beneficial but a fundamental requirement for industrial control-driven systems, motivating the development of foundation models explicitly designed for multivariate industrial environments. Architectural choices are thus the primary driver of generalization across industrial time series settings.

These findings also directly support the central claim of this work regarding cross-site generalization. When compared against baseline models such as Random Forest, LSTM, and TiDE, which require full task-specific training, TSFMs consistently maintain strong performance without any task-specific adaptation. The observed performance degradation of these baselines under regime shift highlights the difficulty of transferring conventionally trained models across sites, suggesting that covariate-aware TSFMs represent a more practical approach for data-scarce industrial settings.

Beyond forecasting, anomaly detection revealed a more complex picture. Task complexity governs the performance ceiling: when faults are gradual and slow-moving, forecasting quality actively decouples from anomaly detection performance, as strong forecasters adapt to degradation and suppress the signals that detection relies on. Unsupervised methods are particularly vulnerable to this, while supervised classifiers can circumvent it when sufficient labeled data is available.

Model embeddings proved consistently informative across all tasks and fault types, strengthening the case for embedding-based anomaly detection pipelines. However, discriminative information within these

embeddings required explicit supervision to extract reliably. Unsupervised approaches such as Isolation Forest yielded inconsistent results, while supervised MLP classifiers achieved strong and consistent performance. This dependency on labeled data represents a practical constraint for industrial deployment, where fault labels are scarce, expensive to obtain, and rarely representative of the full fault space.

Several directions follow naturally from these findings. Field validation on real-world datasets remains an important open avenue, as the controlled nature of the evaluation setting limits conclusions about robustness under noise and sensor drift, conditions often present in live industrial environments. Fine-tuning strategies require investigation: this study evaluated foundation models in a zero-shot setting, leaving performance gains from full fine-tuning, parameter-efficient adaptation, and few-shot approaches unexplored. Beyond forecasting, evaluating the effect of varying context lengths and forecasting horizons on anomaly detection sensitivity represents a natural extension. A more systematic hyperparameter analysis for embedding-based detection methods may further close the performance gap between unsupervised and supervised methods. Finally, as deployment at scale requires simultaneous forecasting across hundreds of sensors, understanding how architectural choices affect inference efficiency remains an open and relevant question.

Ultimately, this work demonstrates that pretrained foundation models represent a viable and often superior alternative to task-specific approaches in industrial time series applications, but realizing their full potential requires careful alignment between model architecture, fault characteristics, and the availability of labeled operational data.

References

Abdalla, R., Samara, H., Perozo, N., Paz, C., & Jaeger, P. (2022, 05). Machine learning approach for predictive maintenance of the electrical submersible pumps (esps). *ACS Omega*, *7*. doi: 10.1021/acsomega.1c05881

Al Harrasi, M., Kazemi, A., & Yousefzadeh, R. (2025). Ensemble learning for prediction of inorganic scale formation: A case study in oman. Retrieved from `https://doi.org/10.1038/s41598-025-05003-2` doi: 10.1038/s41598-025-05003-2

Andrijić, , Bolf, N., Rimac, N., & Brzović, A. (2021, 12). Fouling detection in industrial heat exchanger using number of transfer units method, neural network, and nonlinear finite impulse response models. *Heat Transfer Engineering*, *43*, 1-16. doi: 10.1080/01457632.2021.2016149

Ansari, A. F., Shchur, O., Küken, J., Auer, A., Han, B., Mercado, P., ... Bohlke-Schneider, M. (2025). *Chronos-2: From univariate to universal forecasting.* Retrieved from `https://arxiv.org/abs/2510.15821`

Arango, S. P., Mercado, P., Kapoor, S., Ansari, A. F., Stella, L., Shen, H., ... Rangapuram, S. S. (2025). *Chronosx: Adapting pretrained time series models with exogenous variables.* Retrieved from `https://arxiv.org/abs/2503.12107`

Axelsson, G., & Stefánsson, V. (2003). Sustainable management of geothermal resources.. Retrieved from `https://api.semanticscholar.org/CorpusID:55996236`

Ben Aoun, M. A., Pasquier, P., & Nguyen, A. (2026). Numerical and lstm-based modeling of physical clogging effects in geothermal injection wells. *Geothermics*, *138*, 103661. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0375650526000660` doi: https://doi.org/10.1016/j.geothermics.2026.103661

Bommasani, R., Hudson, D., Adeli, E., Altman, R., Arora, S., Arx, S., ... Liang, P. (2021, 08). *On the opportunities and risks of foundation models.* doi: 10.48550/arXiv.2108.07258

Bondad, K. A. J., & Redoña, B. M. (2026, April). A risk-informed approach to cooling tower maintenance: Root cause analysis and strategic interventions in a geothermal power plant. *Ignatian International Journal for Multidisciplinary Research*, *4*(4). Retrieved from `https://doi.org/10.5281/zenodo.19711365` doi: 10.5281/zenodo.19711365

Boniol, P., Liu, Q., Huang, M., Palpanas, T., & Paparrizos, J. (2024). *Dive into time-series anomaly detection: A decade review.* Retrieved from `https://arxiv.org/abs/2412.20512`

Breiman, L. (2001, October). Random forests. *Mach. Learn.*, *45*(1), 5–32. Retrieved from `https://doi.org/10.1023/A:1010933404324` doi: 10.1023/A:1010933404324

Brown, C. S., Cassidy, N. J., Egan, S. S., & Griffiths, D. (2022). Thermal and economic analysis of heat exchangers as part of a geothermal district heating scheme in the cheshire basin, uk. *Energies*, *15*(6). Retrieved from `https://www.mdpi.com/1996-1073/15/6/1983` doi: 10.3390/en15061983

Chand, S., Subhani, M., Sravani, P., & Ijmtst, E. (2021, 11). The systematic comparision on analysis of parallel flow and counter flow heat exchanger by using cfd and practicle methods. *International Journal for Modern Trends in Science and Technology*, *7*, 153-161. doi: 10.46501/IJMTST0711026

Darban, Z. Z., Webb, G. I., Pan, S., Aggarwal, C., & Salehi, M. (2024, October). Deep learning for time series anomaly detection: A survey. *ACM Computing Surveys*, *57*(1), 1–42. Retrieved from http://dx.doi.org/10.1145/3691338 doi: 10.1145/3691338

Das, A., Faw, M., Sen, R., & Zhou, Y. (2024a). *In-context fine-tuning for time-series foundation models.* Retrieved from https://arxiv.org/abs/2410.24087

Das, A., Kong, W., Leach, A., Mathur, S., Sen, R., & Yu, R. (2024c). *Long-term forecasting with tide: Time-series dense encoder.* Retrieved from https://arxiv.org/abs/2304.08424

Das, A., Kong, W., Sen, R., & Zhou, Y. (2024b). *A decoder-only foundation model for time-series forecasting.* Retrieved from https://arxiv.org/abs/2310.10688

Delibasoglu, I., & Heintz, F. (2024). Time Series Anomaly Detection Leveraging MSE Feedback with AutoEncoder and RNN. In P. Sala, M. Sioutis, & F. Wang (Eds.), *31st international symposium on temporal representation and reasoning (time 2024)* (Vol. 318, pp. 17:1–17:12). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik. Retrieved from https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.TIME.2024.17 doi: 10.4230/LIPIcs.TIME.2024.17

Fakher, S., Khlaifat, A., Hossain, M., & Nameer, H. (2021, 09). Rigorous review of electrical submersible pump failure mechanisms and their mitigation measures. *Journal of Petroleum Exploration and Production Technology*, *11*. doi: 10.1007/s13202-021-01271-6

Gholamalinezhad, H., & Khosravi, H. (2020). *Pooling methods in deep neural networks, a review.* Retrieved from https://arxiv.org/abs/2009.07485

Gneiting, T., & Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, *102*(477), 359–378. Retrieved from https://doi.org/10.1198/016214506000001437 doi: 10.1198/016214506000001437

Hashemi, L., Palochis, D., Poort, J., Octaviano, R., & Omrani, P. S. (2025, 06). *Advancing geothermal operations with digital twin technology: Proactive monitoring and optimization* (Vol. SPE Europe Energy Conference and Exhibition). Retrieved from https://doi.org/10.2118/225592-MS doi: 10.2118/225592-MS

Herzen, J., Lässig, F., Piazzetta, S. G., Neuer, T., Tafti, L., Raille, G., ... Grosch, G. (2022). Darts: User-friendly modern machine learning for time series. *Journal of Machine Learning Research*, *23*(124), 1-6. Retrieved from http://jmlr.org/papers/v23/21-1177.html

Hochreiter, S., & Schmidhuber, J. (1997, 11). Long short-term memory. *Neural Computation*, *9*, 1735-1780. doi: 10.1162/neco.1997.9.8.1735

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359-366. Retrieved from https://www.sciencedirect.com/science/article/pii/0893608089900208 doi: https://doi.org/10.1016/0893-6080(89)90020-8

IEA. (2024). *The future of geothermal energy.* Paris. Retrieved from https://www.iea.org/reports/the-future-of-geothermal-energy (Licence: CC BY 4.0)

Iwata, T., & Kumagai, A. (2020). *Few-shot learning for time-series forecasting.* Retrieved from https://arxiv.org/abs/2009.14379

Kottapalli, S. R. K., Hubli, K., Chandrashekhara, S., Jain, G., Hubli, S., Botla, G., & Doddaiah, R. (2025). *Foundation models for time series: A survey.* Retrieved from https://arxiv.org/abs/2504.04011

Kreuzberger, D., Kühl, N., & Hirschl, S. (2023, 01). Machine learning operations (mlops): Overview, definition, and architecture. *IEEE Access*, *PP*, 1-1. doi: 10.1109/ACCESS.2023.3262138

Kumar, A., & Alam, T. (2025). A review on geothermal energy systems and various approaches to enhance the system's performance. *Energy and Buildings*, *344*, 115962. Retrieved from https://www.sciencedirect.com/science/article/pii/S0378778825006929 doi: https://doi.org/10.1016/j.enbuild.2025.115962

Lee, T., Gottschlich, J., Tatbul, N., Metcalf, E., & Zdonik, S. (2018, 01). Precision and recall for range-based anomaly detection.
doi: 10.48550/arXiv.1801.03175

Liang, Y., Wen, H., Nie, Y., Jiang, Y., Jin, M., Song, D., ... Wen, Q. (2024). Foundation models for time series analysis: A tutorial and survey. In *Proceedings of the 30th acm sigkdd conference on knowledge discovery and data mining* (p. 6555–6565). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/3637528.3671451 doi: 10.1145/3637528.3671451

Liu, F. T., Ting, K., & Zhou, Z.-H. (2009, 01). Isolation forest. In (p. 413 - 422). doi: 10.1109/ICDM.2008.17

Lund, J. (2006, 01). Direct heat utilization of geothermal resources worldwide 2005. *ASEG Extended Abstracts*, *2006*. doi: 10.1071/ASEG2006ab099

Lund, J. W., & Toth, A. N. (2021). Direct utilization of geothermal energy 2020 worldwide review. *Geothermics*, *90*, 101915. Retrieved from https://www.sciencedirect.com/science/article/pii/S0375650520302078 doi: https://doi.org/10.1016/j.geothermics.2020.101915

Marconi, B. (2025, 07). *Time series foundation models for multivariate financial time series forecasting.* doi: 10.48550/arXiv.2507.07296

Meyer, M., Zapata Gonzalez, D., Kaltenpoth, S., & Müller, O. (2025). Benchmarking time series foundation models for short-term household electricity load forecasting. *IEEE Access*, *13*, 218141–218153. Retrieved from http://dx.doi.org/10.1109/ACCESS.2025.3648056 doi: 10.1109/access.2025.3648056

Mulayim, O. B., Quan, P., Han, L., Ouyang, X., Hong, D., Bergés, M., & Srivastava, M. (2025). *Can time-series foundation models perform building energy management tasks?* Retrieved from https://arxiv.org/abs/2506.11250

Nieuwe Warmte Nu. (n.d.). *Resultaten.* Retrieved 2026-06-08, from https://nwn.nu/resultaten/

Nogara, J., & Zarrouk, S. J. (2018). Corrosion in geothermal environment: Part 1: Fluids and their impact. *Renewable and Sustainable Energy Reviews*, *82*, 1333-1346. Retrieved from https://www.sciencedirect.com/science/article/pii/S1364032117310377 doi: https://doi.org/10.1016/j.rser.2017.06.098

Nunes, P., Santos, J., & Rocha, E. (2023). Challenges in predictive maintenance – a review. *CIRP Journal of Manufacturing Science and Technology*, *40*, 53-67. Retrieved from https://www.sciencedirect.com/science/article/pii/S1755581722001742 doi: https://doi.org/10.1016/j.cirpj.2022.11.004

Ogbonnaya, S., & Ajayi, O. (2017, 12). Fouling phenomenon and its effect on heat exchanger: A review. *Frontiers in Heat and Mass Transfer*, *9*. doi: 10.5098/hmt.9.31

Omrani, P. S., & de With, G. (2025). Chapter 9 - production and operation of geothermal systems. In S. Livescu & B. Dindoruk (Eds.), *Geothermal energy engineering* (p. 261-302). Elsevier. Retrieved from https://www.sciencedirect.com/science/article/pii/B9780443216626000025 doi: https://doi.org/10.1016/B978-0-443-21662-6.00002-5

Omrani, P. S., Egberts, P. J., Rijnaarts, H. H., & Shariat Torbaghan, S. (2026). Geothermal plant operation and control under demand uncertainties. *Renewable Energy*, *257*, 124805. Retrieved from https://www.sciencedirect.com/science/article/pii/S0960148125024693 doi: https://doi.org/10.1016/j.renene.2025.124805

Omrani, P. S., Hashemi, L., Poort, J., Schouten, A., Egberts, P. J., Octaviano, R., & Palochis, D. (n.d.). An open digital twin platform for co-simulation and optimization of geothermal plant operations.. Retrieved from https://api.semanticscholar.org/CorpusID:285260662

Omrani, P. S., Poort, J., & Shahmohammadi, S. (2025a). Chapter 11 - artificial intelligence in the geothermal energy systems. In S. Livescu & B. Dindoruk (Eds.), *Geothermal energy engineering* (p. 349-377). Elsevier. Retrieved from https://www.sciencedirect.com/science/article/pii/B9780443216626000049 doi: https://doi.org/10.1016/B978-0-443-21662-6.00004-9

Omrani, P. S., Van der Valk, K., Bos, W., Nizamutdinov, E., Van der Sluijs, L., Eilers, J., ... Van Bergen, F. (2021, 10). *Overview of opportunities and challenges of electrical submersible pumps esp in the geothermal energy production systems* (Vol. SPE Gulf Coast Section Electric Submersible Pumps Symposium). Retrieved from https://doi.org/10.2118/204524-MS doi: 10.2118/204524-MS

Omrani, P. S., Yang, Y., Rijnaarts, H. H., & Torbaghan, S. S. (2025b). Real-time model-based condition monitoring of geothermal systems under uncertainties – case study on electrical submersible pumps. *Geoenergy Science and Engineering*, *249*, 213775. Retrieved from https://www.sciencedirect.com/science/article/pii/S2949891025001332 doi: https://doi.org/10.1016/j.geoen.2025.213775

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., ... Snoek, J. (2019). *Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift.* Retrieved from https://arxiv.org/abs/1906.02530

Pambudi, N. A., Itoi, R., Yamashiro, R., CSS Syah Alam, B. Y., Tusara, L., Jalilinasrabady, S., & Khasani, J. (2015). The behavior of silica in geothermal brine from dieng geothermal power plant, indonesia. *Geothermics*, *54*, 109-114. Retrieved from https://www.sciencedirect.com/science/article/pii/S0375650514001382 doi: https://doi.org/10.1016/j.geothermics.2014.12.003

Park, Y. J., Germain, F., Liu, J., Wang, Y., Koike-Akino, T., Wichern, G., ... Chakrabarty, A. (2025). *Probabilistic forecasting for building energy systems using time-series foundation models.* Retrieved from https://arxiv.org/abs/2506.00630

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

Peng, L., Han, G., Sui, X., Arnold, L. P., Zhu, L., & Shu, J. (2021, 03). Predictive approach to perform fault detection in electrical submersible pump systems. *ACS Omega*, *XXXX*. doi: 10.1021/acsomega.0c05808

Penot, C., Martelo, D., & Paul, S. (2023). Corrosion and scaling in geothermal heat exchangers. *Applied Sciences*, *13*(20). Retrieved from https://www.mdpi.com/2076-3417/13/20/11549 doi: 10.3390/app132011549

Pratap, S., Aranha, A. R., Kumar, D., Malhotra, G., Iyer, A. P. N., & S.S., S. (2025). The fine art of fine-tuning: A structured review of advanced llm fine-tuning techniques. *Natural Language Processing Journal*, *11*, 100144. Retrieved from https://www.sciencedirect.com/science/article/pii/S2949719125000202 doi: https://doi.org/10.1016/j.nlp.2025.100144

Purwaningsih, F. O., & Abdurrahman, G. (2016, February). Geothermal brine, from waste to alternative thermal energy source. In *Proceedings, 41st workshop on geothermal reservoir engineering.* Stanford, California. Retrieved from https://pangea.stanford.edu/ERE/db/IGAstandard/record_detail.php?id=26506

Qin, G., Chen, Z., Liu, Y., Shi, Z., Liu, H., Huang, X., ... Long, M. (2025). *Cora: Covariate-aware adaptation of time series foundation models.* Retrieved from https://arxiv.org/abs/2510.12681

Ryohei Izawa, M. K., Ryosuke Sato. (2021). *Prts: Python library for time series metrics.* Zenodo. Retrieved from https://zenodo.org/record/4428056 doi: 10.5281/ZENODO.4428056

Shannon, D. W. (1975, 01). *Economic impact of corrosion and scaling problems in geothermal energy systems* (Tech. Rep.). Battelle Pacific Northwest Labs., Richland, Wash. (US). Retrieved from https://www.osti.gov/biblio/5122645 doi: 10.2172/5122645

Shchur, O., Ansari, A. F., Turkmen, C., Stella, L., Erickson, N., Guerron, P., ... Wang, Y. (2025). *fev-bench: A realistic benchmark for time series forecasting.* Retrieved from https://arxiv.org/abs/2509.26468

Shetty, P., Lam, T., Songchitruksa, P., Kohar, A., Benslimane, S., Roychoudhury, I., ... Celaya, J. (2025, 11). Architecting asset specific time series foundation model and its applications for asset performance management.. doi: 10.2118/229309-MS

Sun, Y., Wang, F., Zhu, Y., Zhao, W. X., & Mao, J. (2024). *An integrated data processing framework for pretraining foundation models.* Retrieved from https://arxiv.org/abs/2402.16358

Villa, L., & Brusamarello, C. Z. (2025). Application of machine learning in monitoring fouling in heat exchangers in chemical engineering: A systematic review. *The Canadian Journal of Chemical Engineering*, *103*(4), 1786-1801. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1002/cjce.25480 doi: https://doi.org/10.1002/cjce.25480

Wei, Q., Tan, C., Gao, X., Guan, X., & Shi, X. (2024). Research on early warning model of electric submersible pump wells failure based on the fusion of physical constraints and data-driven approach. *Geoenergy Science and Engineering*, *233*, 212489. Retrieved from https://www.sciencedirect.com/science/article/pii/S294989102301076X doi: https://doi.org/10.1016/j.geoen.2023.212489

Whole Building Design Guide (WBDG). (n.d.). *Geothermal energy – direct-use.* https://www.wbdg.org/resources/geothermal-energy-direct-use. (Accessed: 2026-01-29)

Wirth, R., & Hipp, J. (2000, 01). Crisp-dm: Towards a standard process model for data mining. *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*.

Woo, G., Liu, C., Kumar, A., Xiong, C., Savarese, S., & Sahoo, D. (2024). *Unified training of universal time series forecasting transformers.* Retrieved from https://arxiv.org/abs/2402.02592

Yang, P., Chen, J., Wu, L., & Li, S. (2022). Fault identification of electric submersible pumps based on unsupervised and multi-source transfer learning integration. *Sustainability*, *14*(16). Retrieved from https://www.mdpi.com/2071-1050/14/16/9870 doi: 10.3390/su14169870

Zarrouk, S. J., & Moon, H. (2014). Efficiency of geothermal power plants: A worldwide review. *Geothermics*, *51*, 142-153. Retrieved from https://www.sciencedirect.com/science/article/pii/S0375650513001120 doi: https://doi.org/10.1016/j.geothermics.2013.11.001

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., ... He, Q. (2019). A comprehensive survey on

transfer learning. *CoRR*, *abs/1911.02685*. Retrieved from http://arxiv.org/abs/1911.02685

Appendix

A Forecasting Task — Supplementary Results

	Horizon: 24		Horizon: 48		Horizon: 96	
Model	**R^2 (std)**	**RMSE (std)**	**R^2 (std)**	**RMSE (std)**	**R^2 (std)**	**RMSE (std)**
Naive	0.000	0.000	0.000	0.000	0.000	0.000
RandomForest	0.001	0.003	-	-	-	-
LSTM	0.005	0.018	0.026	0.076	0.052	0.120
TiDE	0.000	0.000	0.000	0.000	0.000	0.000
Chronos	0.000	0.000	0.000	0.000	0.000	0.000
Moirai	0.000	0.001	0.001	0.001	0.002	0.002
TimesFM*	0.000	0.000	0.000	0.000	0.000	0.000
Chronos*	0.000	0.000	0.000	0.000	0.000	0.000
Moirai*	0.000	0.001	0.001	0.003	0.001	0.001

Table 8: Standard deviation of forecasting performance across five independent runs. Models marked with * denote univariate forecasting.

Table 8 summarizes the standard deviation of 5 runs across the models. Traditional deep learning (LSTM) is highly unstable, with performance variance scaling sharply as the horizon expands ($RMSE$ standard deviation rises from 0.005 to 0.120). Conversely, foundation models (Chronos, TimesFM) and TiDE exhibit zero or near-zero variance, demonstrating exceptional architectural stability.

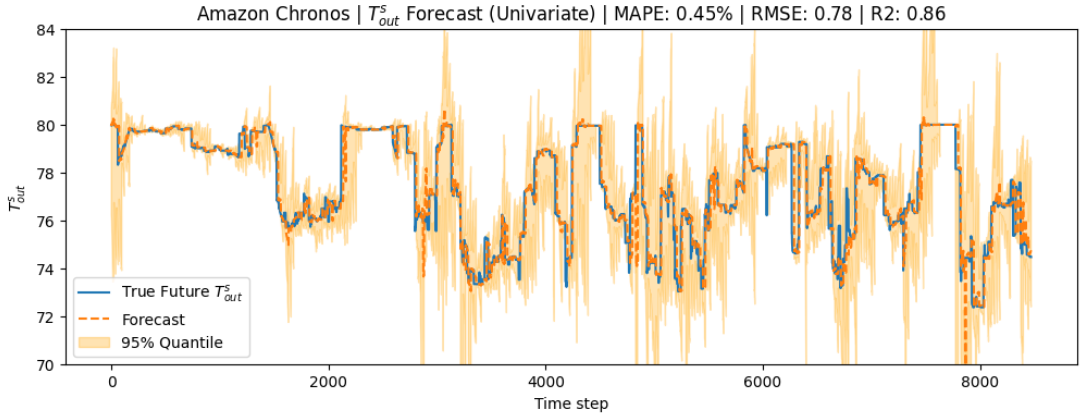
Model	**24 (sec)**	**48 (sec)**	**96 (sec)**
Randomforest	17	–	–
LSTM	1	1	1
TiDE	42	15	8
Chronos	33	17	9
Moirai	196	100	51
TimesFM	82	40	14

Table 9: Execution time comparison across different horizons.

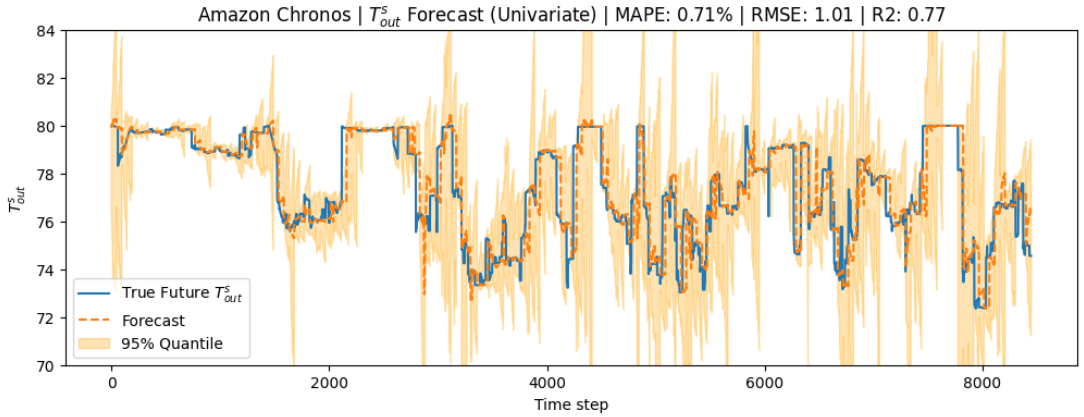
Table 9 summarizes the inference speeds across models. As the forecasting horizon increases from 24 to 96 steps, execution times decrease significantly across nearly all models. This speedup is a direct artifact of dataset constraints. With a fixed total data length, extending the prediction horizon reduces the remaining data available for testing, resulting in fewer valid inference windows and fewer total model evaluations. The single exception is the LSTM, which runs almost instantaneously at a flat 1 second regardless of the horizon.

B Qualitative Forecasts across Models

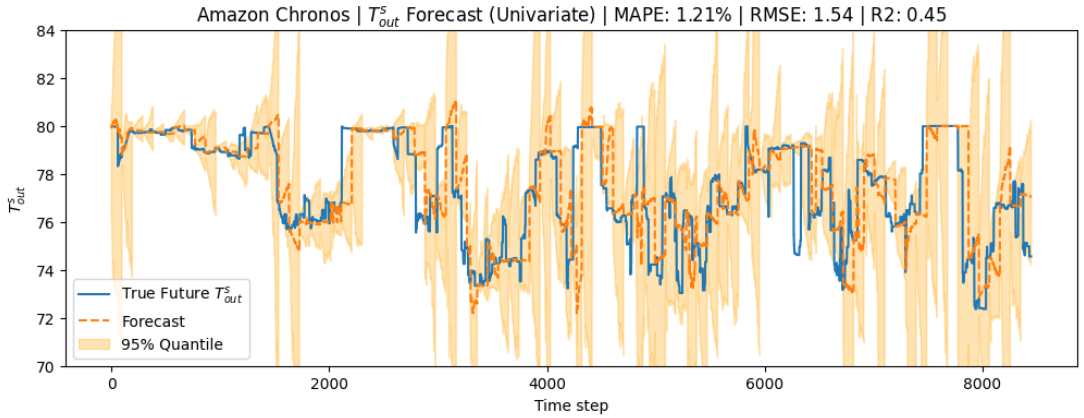
Chronos*



(a) H=24



(b) H=48



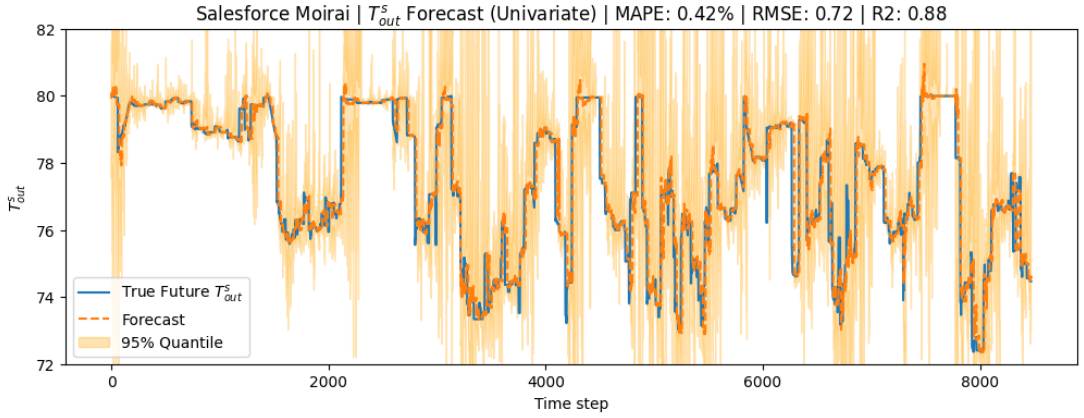
(c) H=96

Figure 16: Chronos (univariate) forecasting across prediction horizons.

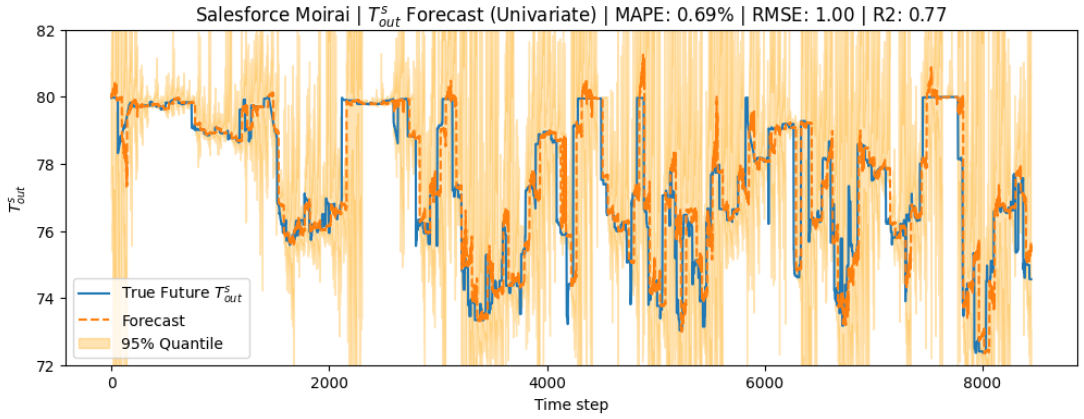
Figure 16 highlights the limitations of univariate forecasting compared to covariate-aware approaches. In the univariate setting, Chronos achieves R^2 scores of 0.86, 0.77, and 0.45 for prediction horizons of 24, 48, and 96 timesteps, respectively. Performance degrades with increasing horizon length and falls below that of the naive baseline (Table 4), aligning more closely with other TSFMs. In contrast, when covariates are incorporated, Chronos demonstrates substantially improved performance. This underscores

the critical role of covariates in accurately modeling control-driven systems.

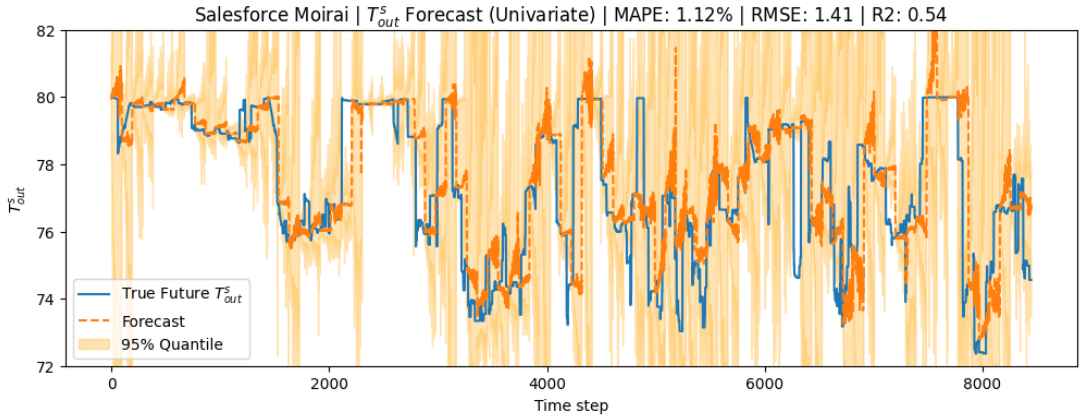
Moirai*



(a) H=24



(b) H=48



(c) H=96

Figure 17: Moirai (univariate) forecasting across prediction horizons.

Figure 17 illustrates Moirai's univariate forecasting performance across increasing prediction horizons. The model achieves comparable performance to its covariate-aware counterpart (Table 4), with minimal degradation observed when excluding covariates. This suggests that Moirai does not effectively leverage additional covariate information in this setting, as the absence of covariates does not significantly impact forecasting accuracy.

C Anomaly Detection — Supplementary Results

Model	Hyperparameter	Values
Isolation Forest	`n_estimators`	100, 300, 500
	`max_samples`	`auto`, 128, 256
	`max_features`	0.5, 1.0
	`bootstrap`	`False`, `True`
MLP	`hidden_layer_sizes`	$(128,), (256,), (256, 64), (512, 128)$
	`activation`	`relu`, `tanh`
	`alpha`	$1e-5, 1e-4, 1e-3$
	`learning_rate_init`	$1e-4, 1e-3$
	`batch_size`	$32, 64, 128$

Table 10: Unified hyperparameter search spaces for Isolation Forest and MLP used in anomaly detection experiments.

Table 10 outlines the grid search configurations utilized for tuning the embedding-based anomaly detection methods, Isolation Forest and Classifier MLP.

C.1 Heat Exchanger Fouling Task

	Forecasting (STD)		Novelty Detection (STD)			Classifier (RF) (STD)		
Model	**R^2**	**RMSE**	**P**	**R**	**F1**	**P**	**R**	**F1**
RandomForest	0.009	0.003	0.056	0.010	0.036	0.000	0.000	0.000
LSTM	0.018	0.022	0.022	0.008	0.014	0.034	0.000	0.013
TiDE	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Chronos	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Moirai	0.001	0.003	0.000	0.002	0.001	0.000	0.003	0.001
TimesFM	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Embedding-Based Anomaly Detection (STD)								
Model			**Isolation Forest**			**Classifier (MLP)**		
LSTM			0.050	0.020	0.024	0.019	0.003	0.010
TiDE			0.000	0.000	0.000	0.000	0.000	0.000
Chronos			0.000	0.000	0.000	0.000	0.000	0.000
Moirai			0.000	0.001	0.000	0.000	0.000	0.000
TimesFM			0.000	0.000	0.000	0.000	0.000	0.000

Table 11: Standard deviation (STD) of forecasting and anomaly detection performance across runs for the heat exchanger task.

Tables 11, 12, and 13 report supplementary details for the heat exchanger task. Table 11 presents the standard deviations of all forecasting and anomaly detection metrics across five independent runs. Table 12 reports inference run-times across models over 34,392 samples, reflecting the computational cost of each approach under identical conditions. Table 13 lists the optimal hyperparameter configurations selected for the Isolation Forest and MLP classifier across all embedding sources following grid search.

Figure 18 presents model forecasts for the heat exchanger fouling task using NTU as the target variable. Differences in model behavior are clearly visible, with some approaches better capturing long-term degradation trends while others exhibit higher variance or lag in response. These qualitative results complement the quantitative evaluation and provide insight into model suitability for anomaly detection in control-driven systems.

Model	Runtime (seconds)
Random Forest	68
LSTM	2
TiDE	115
Chronos	142
Moirai	789
TimeSFM	635

Table 12: Inference Runtimes of Different Models over 34392 samples

Model Space	Hyperparameter	Optimal Value per Foundation Model
Isolation Forest	`bootstrap`	`False` (All models)
	`max_samples`	`auto` (All models)
	`max_features`	0.5 (LSTM, Moirai) 1.0 (TiDE, Chronos, TimesFM)
	`n_estimators`	500 (LSTM, Moirai) 100 (TiDE, Chronos, TimesFM)
Classifier (MLP)	`learning_rate_init`	0.001 (All models)
	`activation`	`relu` (LSTM, TiDE, Chronos) `tanh` (Moirai, TimesFM)
	`alpha`	$1e-4$ (LSTM, TiDE) $1e-5$ (Chronos, TimesFM) $1e-3$ (Moirai)
	`batch_size`	32 (LSTM, TiDE, Moirai) 128 (Chronos) 64 (TimesFM)
	`hidden_layer_sizes`	$(512, 128)$ (LSTM, Moirai) $(256, 64)$ (TiDE, Chronos) $(128,)$ (TimesFM)

Table 13: Optimal hyperparameter configurations for Isolation Forest and MLP classifiers across embedding models for HEX task.

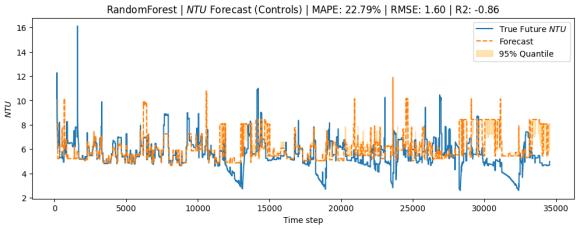
C.2 Electrical Submersible Pump Degradation

Tables 14, 15, and 16 report supplementary details for the electrical submersible pump task. Table 14 presents the standard deviations of all forecasting and anomaly detection metrics across five independent runs. Table 15 reports inference run-times across models over 25,752 samples, reflecting the computational cost of each approach under identical conditions. Table 16 lists the optimal hyperparameter configurations selected for the Isolation Forest and MLP classifier across all embedding sources following grid search.

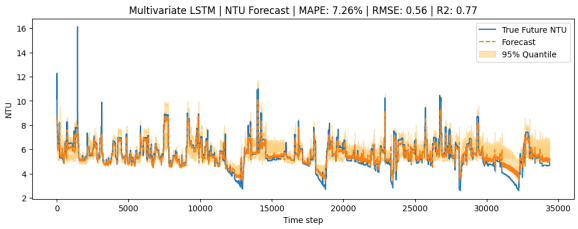
Figure 19 presents model forecasts for electrical submersible pump degradation using pump head as the target variable. Variations in model behavior are evident, with some approaches better capturing gradual degradation patterns while others exhibit delayed responses or increased variability. These qualitative observations complement the quantitative results and provide additional insight into model effectiveness for anomaly detection in industrial systems.

C.3 Filter Clogging

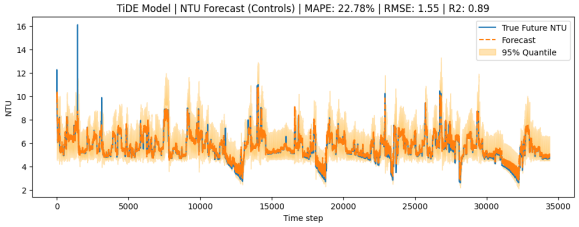
Tables 17, 18, and 19 report supplementary details for the filter task. Table 17 presents the standard deviations of all forecasting and anomaly detection metrics across five independent runs. Table 18 reports inference run-times across models over 17,112 samples, reflecting the computational cost of each approach under identical conditions. Table 19 lists the optimal hyperparameter configurations selected for the Isolation Forest and MLP classifier across all embedding sources following grid search.



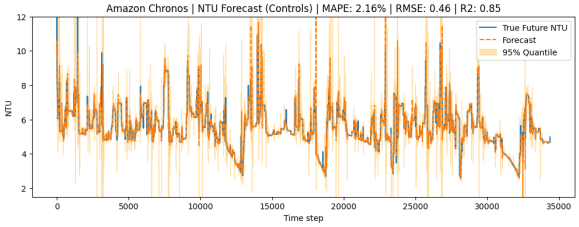
(a) RandomForest



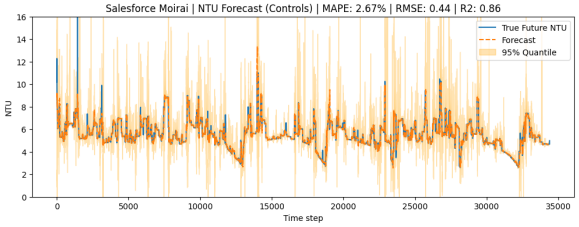
(b) LSTM



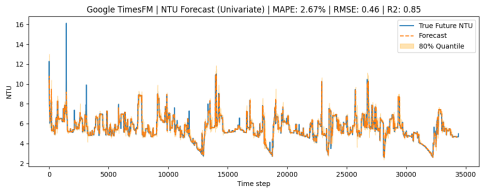
(c) TiDE



(d) Chronos

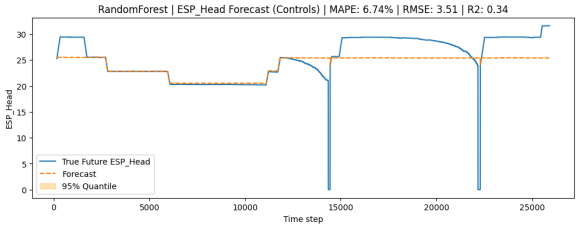


(e) Moirai

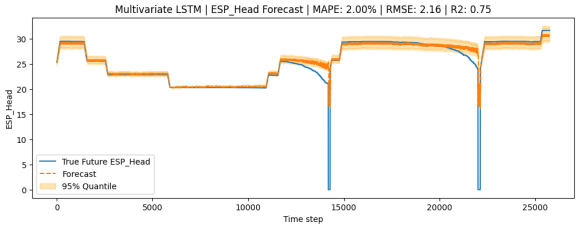


(f) TimesFM

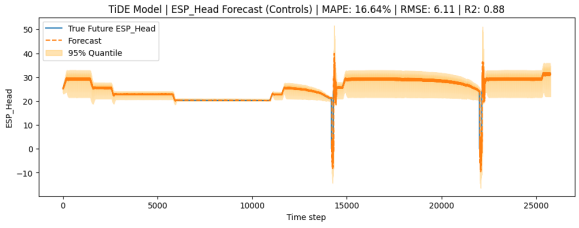
Figure 18: Forecasts for heat exchanger fouling (NTU) across different models.



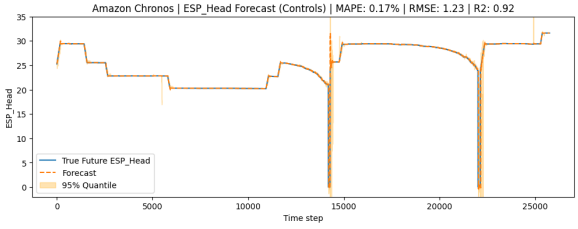
(a) RandomForest



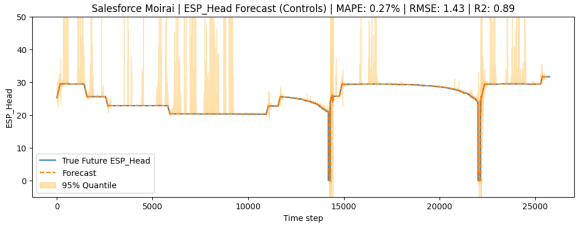
(b) LSTM



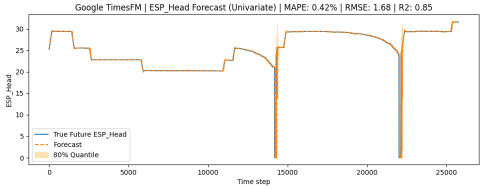
(c) TiDE



(d) Chronos



(e) Moirai



(f) TimesFM

Figure 19: Forecasts for electrical submersible pump degradation (head) across different models.

	Forecasting (STD)		Novelty Detection (STD)			Classifier (RF) (STD)		
Model	**R^2**	**RMSE**	**P**	**R**	**F1**	**P**	**R**	**F1**
RandomForest	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
LSTM	0.016	0.070	0.095	0.021	0.066	0.072	0.000	0.073
TiDE	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Chronos	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Moirai	0.002	0.000	0.001	0.000	0.002	0.001	0.000	0.002
TimesFM	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Embedding-Based Anomaly Detection (STD)								
Model			**Isolation Forest**			**Classifier (MLP)**		
LSTM			0.063	0.004	0.031	0.288	0.006	0.162
TiDE			0.000	0.000	0.000	0.000	0.000	0.000
Chronos			0.000	0.000	0.000	0.000	0.000	0.000
Moirai			0.000	0.000	0.000	0.000	0.000	0.000
TimesFM			0.000	0.000	0.000	0.000	0.000	0.000

Table 14: Standard deviation (STD) of forecasting and anomaly detection performance across runs for the electrical submersible pump task.

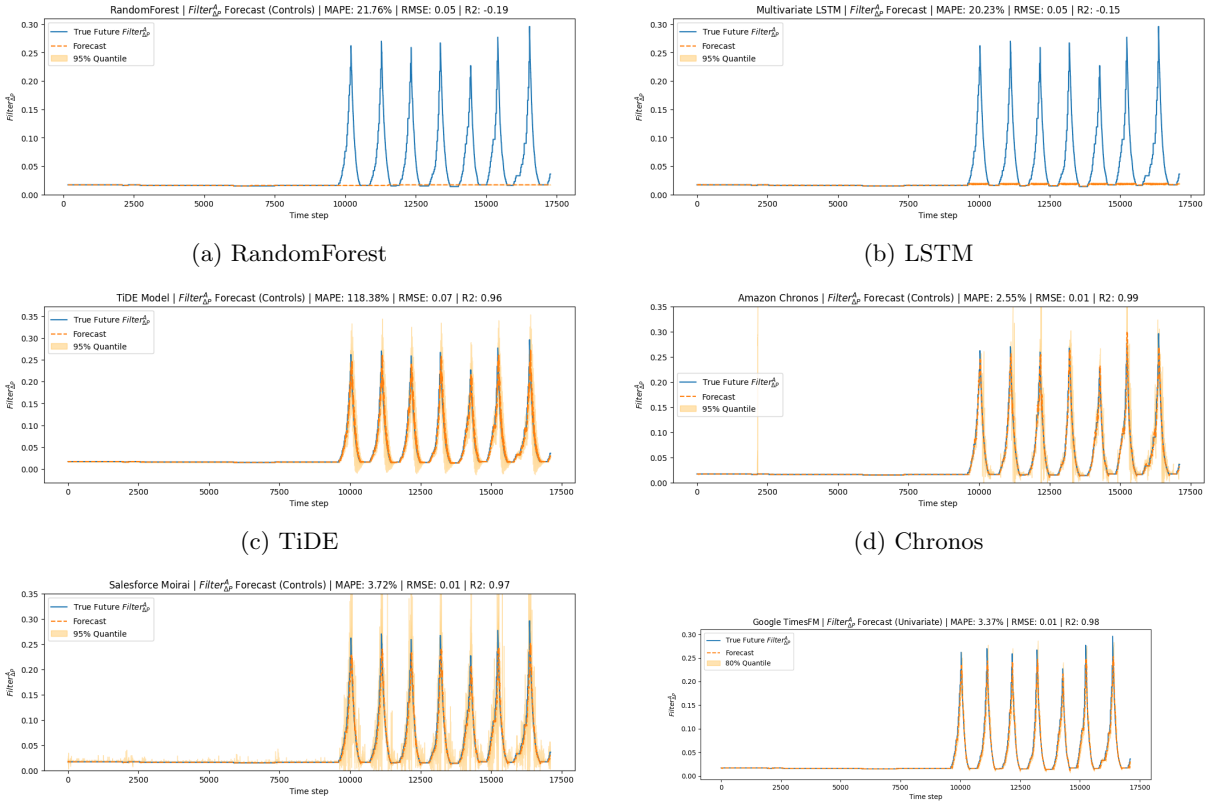
Model	**Runtime (seconds)**
Random Forest	52
LSTM	1
TiDE	86
Chronos	104
Moirai	589
TimeSFM	486

Table 15: Inference Runtimes of Different Models over 25752 samples

Figure 20 presents model forecasts for the filter clogging task using pressure drop as the target variable. Differences in model behavior are evident, with some approaches more accurately tracking gradual increases associated with clogging, while others stick to the training distribution, ignoring the distribution shift entirely. These qualitative results complement the quantitative evaluation and provide further insight into model suitability for anomaly detection in control-driven systems.

Model Space	Hyperparameter	Optimal Value per Foundation Model
Isolation Forest	`bootstrap`	`False` (All models)
	`max_samples`	`auto` (All models)
	`max_features`	0.5 (LSTM, TiDE, TimesFM) 1.0 (Chronos, Moirai)
	`n_estimators`	300 (LSTM) 100 (TiDE, Moirai, TimesFM) 500 (Chronos)
Classifier (MLP)	`learning_rate_init`	0.001 (All models)
	`alpha`	$1e-5$ (All models)
	`activation`	`tanh` (TiDE, Chronos, TimesFM) `relu` (Moirai)
	`batch_size`	32 (TiDE, Chronos, Moirai) 64 (TimesFM)
	`hidden_layer_sizes`	$(512, 128)$ (TiDE) $(256,)$ (Chronos) $(256, 64)$ (Moirai) $(128,)$ (TimesFM)

Table 16: Optimal hyperparameter configurations for Isolation Forest and MLP classifiers across embedding models for ESP task.



(a) RandomForest

(b) LSTM

(c) TiDE

(d) Chronos

(e) Moirai

(f) TimesFM

Figure 20: Forecasts for filter clogging (pressure drop) across different models.

	Forecasting (STD)		Novelty Detection (STD)			Classifier (RF) (STD)		
Model	**R^2**	**RMSE**	**P**	**R**	**F1**	**P**	**R**	**F1**
RandomForest	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
LSTM	0.008	0.000	0.053	0.074	0.074	0.022	0.022	0.017
TiDE	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Chronos	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Moirai	0.001	0.000	0.000	0.001	0.000	0.000	0.001	0.000
TimesFM	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Embedding-Based Anomaly Detection (STD)								
Model			**Isolation Forest**			**Classifier (MLP)**		
LSTM			0.001	0.000	0.003	0.000	0.001	0.000
TiDE			0.000	0.000	0.000	0.000	0.000	0.000
Chronos			0.000	0.000	0.000	0.000	0.000	0.000
Moirai			0.000	0.000	0.000	0.000	0.000	0.000
TimesFM			0.000	0.000	0.000	0.000	0.000	0.000

Table 17: Standard deviation (STD) of forecasting and anomaly detection performance across runs for the filter task.

Model	**Runtime (seconds)**
Random Forest	63
LSTM	1
TiDE	57
Chronos	69
Moirai	384
TimeSFM	257

Table 18: Inference Runtimes of Different Models over 17112 samples

Model Space	**Hyperparameter**	**Optimal Value per Foundation Model**
Isolation Forest	`bootstrap`	`False` (All models)
	`max_samples`	`auto` (All models)
	`max_features`	0.5 (LSTM, Moirai, TimesFM) 1.0 (TiDE, Chronos)
	`n_estimators`	100 (LSTM) 300 (Moirai) 500 (TiDE, Chronos, TimesFM)
Classifier (MLP)	`learning_rate_init`	0.001 (All models)
	`alpha`	$1e-5$ (All models)
	`activation`	`relu` (LSTM, Chronos, Moirai) `tanh` (TiDE, TimesFM)
	`batch_size`	32 (LSTM, TiDE, TimesFM) 64 (Moirai) 128 (Chronos)
	`hidden_layer_sizes`	$(128,)$ (LSTM, Chronos, TimesFM) $(512, 128)$ (TiDE, Moirai)

Table 19: Optimal hyperparameter configurations for Isolation Forest and MLP classifiers across embedding models for filter task.

4

Conclusion and Future Work

In this thesis, we first established the critical context: the urgent climate issues overwhelming our planet, and how current attempts to replace fossil fuels are limited by the intermittency of other renewable energy sources. This directly led us to the core focus of this work: geothermal energy systems. While these systems are crucial for continuous power, they are severely constrained by data scarcity, rare fault events, and a major limitation in current machine learning workflows: poor cross-site generalizability. This specific bottleneck motivated the central question of this thesis: whether Time Series Foundation Models can serve as a practical zero-shot solution for condition monitoring in these systems.

Evaluating TSFMs in this domain proved to be a unique challenge. Most prior research on TSFMs has focused on univariate settings, where model performance is driven largely by historical trends and seasonality. Geothermal systems, however, are covariate-driven: operator controls influence the bulk of the observed sensor behavior, rendering univariate approaches ill-suited for this class of problems. This thesis showed that recent architectural advancements, specifically mechanisms that explicitly integrate exogenous covariates, are critical in bridging this gap, enabling the generalization capability that TSFMs were designed to offer.

TSFMs were evaluated across two axes: forecasting and anomaly detection. In forecasting, performance proved highly dependent on covariate integration. Chronos outperformed all baselines and competing foundation models by substantial margins, achieving 22–35% reductions in RMSE across forecasting horizons. This result carries two important conclusions. First, covariate-aware TSFMs are strong zero-shot forecasters, capable of outperforming models that received full task-specific training. Second, architectural choices are decisive: deactivating Chronos' covariate integration caused its performance to drop below the trained baselines, a pattern consistent with Moirai and TimesFM, which were not designed with explicit covariate handling and underperformed accordingly. Together, these findings demonstrate that architectural capability is a primary driver of a foundation model's zero-shot utility in complex physical systems.

Anomaly detection revealed a more complex picture, evaluated through two different approaches: forecasting-based and embedding-based detection. Forecasting-based novelty detection exposed a fundamental tension: strong forecasters such as Chronos adapt smoothly to gradual degradation, suppressing anomaly flags. Forecasting accuracy and anomaly detection sensitivity are therefore decoupled for gradual degradations, an important practical limitation. Pairing forecast outputs with a supervised classifier partially addressed this, but performance was dependent on the availability of sufficient labeled fault data and the classifier's ability to generalize across the fault types. When these conditions were met, strong forecasters benefited; when they were not, performance degraded across the board.

Embedding-based detection provided further insight. Isolation Forest yielded inconsistent results across tasks, revealing that forecast quality does not imply embedding separability. Chronos, the strongest forecaster, produced the highest-dimensional embeddings, which proved difficult for an unsupervised method to separate effectively. The supervised MLP classifier on embeddings was the only approach to yield consistent and reliable performance across all tasks and fault types. This confirmed that TSFM

embeddings are information-rich representations of system state, but the discriminative information they contain requires supervised extraction to be reliably utilized. Four conclusions follow from these findings: covariate-aware TSFMs are effective zero-shot forecasters; architectural design is the primary driver in control-driven environments; task complexity governs anomaly detection performance and decouples it from forecasting quality; and TSFM embeddings encode meaningful information that enables reliable anomaly detection when paired with a supervised classifier.

The overarching research question of this thesis has been answered. Covariate-aware TSFMs are a practical and effective zero-shot solution for condition monitoring in geothermal energy systems, outperforming conventional task-specific baselines that required full supervised training. This positions TSFMs as a viable foundation for condition monitoring not only in geothermal operations, but across complex industrial environments and renewable energy infrastructures that are increasingly important in a sustainable, non-fossil fuel future.

Several directions for future work follow naturally from these findings. Fine-tuning strategies require investigation, as this thesis evaluated models exclusively in a zero-shot setting and the additional performance gains from fine-tuning remain unquantified. Broader application across other covariate-driven industrial domains would solidify the findings presented here. Finally, more extensive testing across varying context lengths, forecasting horizons, and fault types would strengthen conclusions about the robustness of TSFMs under the conditions encountered in live industrial deployments.

Looking ahead, it is clear that Time Series Foundation Models will only continue to evolve. By demonstrating that the deliberate integration of covariates is what allows these models to outperform standard baselines, this thesis underscores that architectural design is the primary driver of success in control-driven industrial environments. As these architectural advancements progress, TSFMs will become increasingly robust, eventually paving the way for full-scale industrial adoption, reducing the reliance on traditional, fragmented machine learning workflows. Ultimately, the future of our energy infrastructure is in our hands. Accelerating the transition to sustainable energy is a necessity for the continuity of our species, and leveraging AI to complement and enhance these systems is our most powerful tool to ensure a safer, more efficient, and secure planet.

References

[1] Fahad Radhi Alharbi and Denes Csala. “A Seasonal Autoregressive Integrated Moving Average with Exogenous Factors (SARIMAX) Forecasting Model-Based Time Series Approach”. In: *Inventions* 7.4 (2022). ISSN: 2411-5134. DOI: 10.3390/inventions7040094. URL: https://www.mdpi.com/2411-5134/7/4/94.

[2] Abdul Fatir Ansari et al. *Chronos-2: From Univariate to Universal Forecasting*. 2025. arXiv: 2510.15821 [cs.LG]. URL: https://arxiv.org/abs/2510.15821.

[3] Mary Asare-Addo. “Wind and solar energy intermittency: The silver lining”. In: *Results in Engineering* 29 (2026), p. 108275. ISSN: 2590-1230. DOI: https://doi.org/10.1016/j.rineng.2025.108275. URL: https://www.sciencedirect.com/science/article/pii/S259012302504321X.

[4] Abdelhakim Benechehab et al. *AdaPTS: Adapting Univariate Foundation Models to Probabilistic Multivariate Time Series Forecasting*. 2025. arXiv: 2502.10235 [stat.ML]. URL: https://arxiv.org/abs/2502.10235.

[5] Rohan Best and Paul J. Burke. “Adoption of solar and wind energy: The roles of carbon pricing and aggregate policy support”. In: *Energy Policy* 118 (2018), pp. 404–417. ISSN: 0301-4215. DOI: https://doi.org/10.1016/j.enpol.2018.03.050. URL: https://www.sciencedirect.com/science/article/pii/S0301421518301848.

[6] Rishi Bommasani et al. *On the Opportunities and Risks of Foundation Models*. 2022. arXiv: 2108.07258 [cs.LG]. URL: https://arxiv.org/abs/2108.07258.

[7] Tom B. Brown et al. *Language Models are Few-Shot Learners*. 2020. arXiv: 2005.14165 [cs.CL]. URL: https://arxiv.org/abs/2005.14165.

[8] Xi Chen et al. *Autoregressive-Model-Based Methods for Online Time Series Prediction with Missing Values: an Experimental Evaluation*. 2019. arXiv: 1908.06729 [stat.ML]. URL: https://arxiv.org/abs/1908.06729.

[9] Abhimanyu Das et al. *A decoder-only foundation model for time-series forecasting*. 2024. arXiv: 2310.10688 [cs.CL]. URL: https://arxiv.org/abs/2310.10688.

[10] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL]. URL: https://arxiv.org/abs/1810.04805.

[11] Annika Eberle et al. *Systematic Review of Life Cycle Greenhouse Gas Emissions from Geothermal Electricity*. American English. Other. 2017. DOI: 10.2172/1398245.

[12] Tawfik Elshehabi and Mohammad Alfehaid. “Sustainable Geothermal Energy: A Review of Challenges and Opportunities in Deep Wells and Shallow Heat Pumps for Transitioning Professionals”. In: *Energies* 18.4 (2025). ISSN: 1996-1073. DOI: 10.3390/en18040811. URL: https://www.mdpi.com/1996-1073/18/4/811.

[13] Benyamin Ghojogh and Ali Ghodsi. *Recurrent Neural Networks and Long Short-Term Memory Networks: Tutorial and Survey*. 2023. arXiv: 2304.11461 [cs.LG]. URL: https://arxiv.org/abs/2304.11461.

[14] Fynn V. Hackstein and Reinhard Madlener. “Sustainable operation of geothermal power plants: why economics matters”. In: *Geothermal Energy* 9.1 (2021). Published 15 March 2021, p. 10. ISSN: 2195-9706. DOI: 10.1186/s40517-021-00183-2. URL: https://doi.org/10.1186/s40517-021-00183-2.

[15] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9 (Nov. 1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[16] Jeremy Howard and Sebastian Ruder. *Universal Language Model Fine-tuning for Text Classification*. 2018. arXiv: 1801.06146 [cs.CL]. URL: https://arxiv.org/abs/1801.06146.

[17] Juan Infante-Amate, Emiliano Travieso, and Eduardo Aguilera. “The history of a + 3 °C future: Global and regional drivers of greenhouse gas emissions (1820–2050)”. In: *Global Environmental Change* 92 (2025), p. 103009. ISSN: 0959-3780. DOI: https://doi.org/10.1016/j.gloenvcha.2025.103009. URL: https://www.sciencedirect.com/science/article/pii/S0959378025000469.

[18] International Energy Agency (IEA). *Global Energy Review 2026*. Licence: CC BY 4.0. Paris: IEA, 2026. URL: https://www.iea.org/reports/global-energy-review-2026.

[19] Jared Kaplan et al. *Scaling Laws for Neural Language Models*. 2020. arXiv: 2001.08361 [cs.LG]. URL: https://arxiv.org/abs/2001.08361.

[20] Mukhtar A. Kassem and Andrea Moscariello. “Geothermal energy: A sustainable and cost-effective alternative for clean energy production and climate change mitigation”. In: *Sustainable Futures* 10 (2025), p. 101247. ISSN: 2666-1888. DOI: https://doi.org/10.1016/j.sftr.2025.101247. URL: https://www.sciencedirect.com/science/article/pii/S2666188825008081.

[21] Feyza Duman Keles, Pruthuvi Mahesakya Wijewardena, and Chinmay Hegde. *On The Computational Complexity of Self-Attention*. 2022. arXiv: 2209.04881 [cs.LG]. URL: https://arxiv.org/abs/2209.04881.

[22] Salman Khalid, Muhammad Muzammil Azad, and Heung Soo Kim. “A Generalized Autonomous Power Plant Fault Detection Model Using Deep Feature Extraction and Ensemble Machine Learning”. In: *Mathematics* 13.3 (2025). ISSN: 2227-7390. DOI: 10.3390/math13030342. URL: https://www.mdpi.com/2227-7390/13/3/342.

[23] Taesung Kim et al. “Reversible Instance Normalization for Accurate Time-Series Forecasting against Distribution Shift”. In: *International Conference on Learning Representations*. 2022. URL: https://openreview.net/forum?id=cGDAkQo1C0p.

[24] Dominik Kreuzberger, Niklas Kühl, and Sebastian Hirschl. “Machine Learning Operations (MLOps): Overview, Definition, and Architecture”. In: *IEEE Access* PP (Jan. 2023), pp. 1–1. DOI: 10.1109/ACCESS.2023.3262138.

[25] Le-Hang Le. “Time series analysis and applications in data analysis, forecasting and prediction”. In: *HPU2 Journal of Science: Natural Sciences and Technology* 3 (Apr. 2024), pp. 20–29. DOI: 10.56764/hpu2.jos.2024.3.1.20-29.

[26] Yuxuan Liang et al. “Foundation Models for Time Series Analysis: A Tutorial and Survey”. In: *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. KDD ’24. Barcelona, Spain: Association for Computing Machinery, 2024, pp. 6555–6565. ISBN: 9798400704901. DOI: 10.1145/3637528.3671451. URL: https://doi.org/10.1145/3637528.3671451.

[27] Bryan Lim et al. *Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting*. 2020. arXiv: 1912.09363 [stat.ML]. URL: https://arxiv.org/abs/1912.09363.

[28] Xiao Liu et al. “Self-supervised Learning: Generative or Contrastive”. In: *IEEE Transactions on Knowledge and Data Engineering* (2021), pp. 1–1. ISSN: 2326-3865. DOI: 10.1109/tkde.2021.3090866. URL: http://dx.doi.org/10.1109/TKDE.2021.3090866.

[29] Camelia Maican et al. “Review of Fault Detection and Diagnosis Methods in Power Plants: Algorithms, Architectures, and Trends”. In: *Applied Sciences* 15 (June 2025), p. 6334. DOI: 10.3390/app15116334.

[30] Takashi Morita. *Positional Encoding Helps Recurrent Neural Networks Handle a Large Vocabulary*. 2024. arXiv: 2402.00236 [cs.LG]. URL: https://arxiv.org/abs/2402.00236.

[31] Yuqi Nie et al. *A Time Series is Worth 64 Words: Long-term Forecasting with Transformers*. 2023. arXiv: 2211.14730 [cs.LG]. URL: https://arxiv.org/abs/2211.14730.

[32] Pejman Shoeibi Omrani and Govert de With. “Chapter 9 - Production and operation of geothermal systems”. In: *Geothermal Energy Engineering*. Ed. by Silviu Livescu and Birol Dindoruk. Elsevier, 2025, pp. 261–302. ISBN: 978-0-443-21662-6. DOI: https://doi.org/10.1016/B978-0-443-21662-6.00002-5. URL: https://www.sciencedirect.com/science/article/pii/B9780443216626000025.

[33] Ayomide Oyemaja. “Moving Averages in Time Series Analysis: Understanding Trends Forecasting.” In: (Mar. 2024).

[34] Corentin Penot, David Martelo, and Shiladitya Paul. “Corrosion and Scaling in Geothermal Heat Exchangers”. In: *Applied Sciences* 13.20 (2023). ISSN: 2076-3417. DOI: 10.3390/app132011549. URL: https://www.mdpi.com/2076-3417/13/20/11549.

[35] Andres Potapczynski et al. *Time-Aware Prior Fitted Networks for Zero-Shot Forecasting with Exogenous Variables*. 2026. arXiv: 2603.15802 [cs.LG]. URL: https://arxiv.org/abs/2603.15802.

[36] Alec Radford and Karthik Narasimhan. “Improving Language Understanding by Generative Pre-Training”. In: 2018. URL: https://api.semanticscholar.org/CorpusID:49313245.

[37] Alec Radford et al. “Language Models are Unsupervised Multitask Learners”. In: *OpenAI* (2019). Accessed: 2024-11-15. URL: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

[38] Sebastian Raschka. *Understanding and Coding the Self-Attention Mechanism of Large Language Models From Scratch*. Accessed: 2026-06-07. Feb. 2023. URL: https://sebastianraschka.com/blog/2023/self-attention-from-scratch.html.

[39] William J Ripple et al. “The 2025 state of the climate report: a planet on the brink”. In: *BioScience* 75.12 (Dec. 2025), pp. 1016–1027. ISSN: 1525-3244. DOI: 10.1093/biosci/biaf149. eprint: https://academic.oup.com/bioscience/article-pdf/75/12/1016/64954020/biaf149.pdf. URL: https://doi.org/10.1093/biosci/biaf149.

[40] Hannah Ritchie and Pablo Rosado. “Fossil fuels”. In: *Our World in Data* (2017). https://ourworldindata.org/fossil-fuels.

[41] Robin M. Schmidt. *Recurrent Neural Networks (RNNs): A gentle Introduction and Overview*. 2019. arXiv: 1912.05911 [cs.LG]. URL: https://arxiv.org/abs/1912.05911.

[42] Rico Sennrich, Barry Haddow, and Alexandra Birch. *Neural Machine Translation of Rare Words with Subword Units*. 2016. arXiv: 1508.07909 [cs.CL]. URL: https://arxiv.org/abs/1508.07909.

[43] D W Shannon. *Economic impact of corrosion and scaling problems in geothermal energy systems*. Tech. rep. Battelle Pacific Northwest Labs., Richland, Wash. (US), Jan. 1975. DOI: 10.2172/5122645. URL: https://www.osti.gov/biblio/5122645.

[44] Sima Siami-Namini and Akbar Siami Namin. *Forecasting Economics and Financial Time Series: ARIMA vs. LSTM*. 2018. arXiv: 1803.06386 [cs.LG]. URL: https://arxiv.org/abs/1803.06386.

[45] Pinar Sungu Isiacik et al. “An Empirical Evaluation of Foundation Models for Multivariate Time Series Classification”. In: June 2025.

[46] Granville Tunnicliffe Wilson. “Time Series Analysis: Forecasting and Control,5th Edition, by George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel and Greta M. Ljung, 2015. Published by John Wiley and Sons Inc., Hoboken, New Jersey, pp. 712. ISBN: 978-1-118-67502-1”. In: *Journal of Time Series Analysis* 37 (Mar. 2016), n/a–n/a. DOI: 10.1111/jtsa.12194.

[47] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

[48] Gerald Woo et al. *Unified Training of Universal Time Series Forecasting Transformers*. 2024. arXiv: 2402.02592 [cs.LG]. URL: https://arxiv.org/abs/2402.02592.

[49] Haixu Wu et al. *Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting*. 2022. arXiv: 2106.13008 [cs.LG]. URL: https://arxiv.org/abs/2106.13008.

[50] Congxi Xiao et al. *TimeFound: A Foundation Model for Time Series Forecasting*. 2025. arXiv: 2503.04118 [cs.LG]. URL: https://arxiv.org/abs/2503.04118.

[51] Jingjing Xu et al. *Understanding and Improving Layer Normalization*. 2019. arXiv: 1911.07013 [cs.LG]. URL: https://arxiv.org/abs/1911.07013.

[52] Peng Yan et al. “A Comprehensive Survey of Deep Transfer Learning for Anomaly Detection in Industrial Time Series: Methods, Applications, and Directions”. In: *IEEE Access* 12 (2024), pp. 3768–3789. ISSN: 2169-3536. DOI: 10.1109/access.2023.3349132. URL: http://dx.doi.org/10.1109/ACCESS.2023.3349132.

[53] Peihao Yang et al. “Fault Identification of Electric Submersible Pumps Based on Unsupervised and Multi-Source Transfer Learning Integration”. In: *Sustainability* 14.16 (2022). ISSN: 2071-1050. DOI: 10.3390/su14169870. URL: https://www.mdpi.com/2071-1050/14/16/9870.

[54] Nita Yodo et al. “Condition-based monitoring as a robust strategy towards sustainable and resilient multi-energy infrastructure systems”. In: *Sustainable and Resilient Infrastructure* 8.sup1 (2023), pp. 170–189. DOI: 10.1080/23789689.2022.2134648. eprint: https://doi.org/10.1080/23789689.2022.2134648. URL: https://doi.org/10.1080/23789689.2022.2134648.

[55] Ailing Zeng et al. *Are Transformers Effective for Time Series Forecasting?* May 2022. DOI: 10.48550/arXiv.2205.13504.

[56] Zhihua Zhang and John Moore. “Autoregressive Moving Average Models”. In: Dec. 2015, pp. 239–290. ISBN: 9780128000663. DOI: 10.1016/B978-0-12-800066-3.00008-5.

[57] Ce Zhou et al. *A Comprehensive Survey on Pretrained Foundation Models: A History from BERT to ChatGPT*. 2023. arXiv: 2302.09419 [cs.AI]. URL: https://arxiv.org/abs/2302.09419.

[58] Haoyi Zhou et al. *Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting*. 2021. arXiv: 2012.07436 [cs.LG]. URL: https://arxiv.org/abs/2012.07436.

[59] Fuzhen Zhuang et al. *A Comprehensive Survey on Transfer Learning*. 2020. arXiv: 1911.02685 [cs.LG]. URL: https://arxiv.org/abs/1911.02685.

A

Declaration of Generative AI Usage

In accordance with the University's academic integrity guidelines, this appendix documents the generative AI tools used during the research, development, and writing of this thesis. I retain full intellectual ownership and responsibility for all content, including text, code, methodologies, and conclusions. AI tools were used only as support, and all outputs were reviewed, verified, and edited by me.

A.1. Literature Support

AI tools were used to assist in searching for relevant scientific literature to support specific claims or to evaluate their validity.

- **Tools Used:** ChatGPT and Copilot
- **My Oversight:** I provided claims to the tools and requested supporting references. All suggested sources were independently verified by manually locating and reviewing the original papers. Given the possibility of hallucinated or incorrect references, all literature was carefully checked before inclusion in the thesis.

A.2. Programming and Data Analysis Support

AI tools were used to assist in implementing parts of the framework and in understanding existing codebases.

- **Tools Used:** Copilot
- **My Oversight:** While I primarily developed the code independently, AI tools were occasionally used to generate code and to help understand complex codebases. All generated code was reviewed, understood, and adapted before use.

A.3. Text Refinement Support

AI tools were used to assist in refining and improving the written content in the thesis.

- **Tools Used:** ChatGPT, Claude, Copilot, and Gemini
- **My Oversight:** I drafted the core ideas and conclusions myself, and used AI tools to improve structure, narrative flow and academic tone, while ensuring that all intellectual content remained my own.

Overall, AI tools were used to enhance my work, not replace independent thinking.