Degree project

UNIVERSITY OF SKÖVDE

Level-Based Unsupervised Anomaly Detection for Industrial Thermocouple Sensor Data in Stainless Steel Furnaces

Master Degree Project in Informatics with a specialization in Data Science

Second Cycle 15 credits

Spring term 2025

Student: Ebrahim Golriz

Supervisor: Alexander Karlsson

Examiner: Nikolaos Kourentzes

Abstract

In stainless steel manufacturing, monitoring accurate temperature is critical for maintaining product quality and process efficiency. Thermocouples which are placed across furnace zones generate sensor data that must be continuously analyzed for early detection of process deviations. This thesis presents a level-based, unsupervised anomaly detection system developed in collaboration with *Outokumpu*, the largest producer of stainless steel in Europe and the second largest producer in the Americas, which aims to identify anomalies in thermocouple and flow sensor data without having labeled training examples.

The proposed system operates across multiple levels: detecting anomalies at the overall furnace level, narrowing down to specific furnace zones, and finally identifying unusual patterns at the individual thermocouple level. This layered approach allows for both broad anomaly detection and detailed diagnostics, which helps to identify the location and potential root cause of abnormal behavior.

This system provides enhanced operational awareness in industrial furnace operations by enabling more targeted investigation of deviations and faster response times. This capability helps with proactive maintenance strategies. The methodology is scalable and data-driven, which improves reliability and efficiency in continuous steel processing.

Table of Contents

1 Introduction

In the stainless steel manufacturing industry, precision in process control is essential to ensuring high product quality, operational efficiency, and energy optimization. One of the most critical variables in this process is temperature, which is continuously monitored using thermocouples, sensors that measure temperature, placed across various zones in industrial furnaces. These sensors provide temperature data that reflect the thermal dynamics of the furnace over time. However, interpreting this data over time to detect subtle but meaningful deviations remains a major challenge.

Traditionally, anomaly detection in such environments relies on expert knowledge, static threshold rules (for instance having a threshold for maximum and minimum value for each thermocouple), or supervised learning methods that require labeled data. However, in real industrial settings, labelled anomaly data is often incomplete, or entirely unavailable, and thresholds can be too simple to capture the nuanced behaviors of complex systems. Moreover, problems in one part of the furnace can be overlooked when only aggregated data is analyzed, which leads to delayed detection of localized issues.

1.1 Problem definition

Identifying the unusual behavior of thermocouples early, before they trigger traditional alarms, is important as it helps finding the issue that if found later, might cause more harm. Modern furnaces have a dense network of sensors, such as thermocouples, distributed across various operational zones. These sensors generate vast streams of multivariate time-series data, which capture the complex thermal dynamics and interdependencies within the furnace environment.

The operational team that work on the furnaces, felt the absence of a dedicated system for proactive thermocouple anomaly detection and the lack of historically labeled data categorizing sensor readings as normal or anomalous. The unavailability of labeled data made supervised learning approaches for fault detection impractical. The high dimensionality and volume of sensor data (including various thermocouple readings like regulation, safety, and wall temperatures per zone, alongside fuel flows and strip speed) further highlighted the need for an intelligent, automated system to continuously monitor furnace health and sensor performance.

The primary aim of this thesis is to design, implement, and evaluate a level-based, unsupervised anomaly detection system to enhance the monitoring of thermocouple readings and the overall thermal process stability across six zones of two furnaces. The system aims to provide early and interpretable detection of anomalies related to individual thermocouples and broader furnace/zone operations, using only historical operational data without relying on pre-existing anomaly labels.

This research aims to address the following key questions:

- Can a hierarchical autoencoder-based system detect anomalies at the furnace and zone levels in unlabeled multivariate time-series data?
- Can individual thermocouple-level regression models provide localized diagnostic insight into detected anomalies?
- What are the strengths and limitations of combining reconstruction-based and prediction-based methods in this industrial context?

To achieve the stated aim, the following objectives have been defined:

- Develop unsupervised anomaly detection models: To design and train autoencoder neural networks for each of the two primary furnaces and each of their operational zones (2 zones in furnace 1 and 4 zones in furnace 2). These models will learn the normal patterns from historical data considering fuel flows (total and split over/under where applicable), diverse thermocouple readings, calculated average steel temperatures, and strip speed.
- Propose an implementation prototype of a level-based detection framework: to construct a multi-level system where furnace-level autoencoders provide an initial assessment. If a furnace-level anomaly is detected, zone-level autoencoders corresponding to the zones within that furnace are then interrogated to localize the source of the irregularity.
- Develop regression models for thermocouples: To build and integrate zone-specific ridge regression models to predict the readings of each individual thermocouple based on other relevant sensor data within its respective zone (such as other TCs in that zone, fuel flows, average steel temperature, and strip speed). Significant deviations between predicted and actual values will serve as a diagnostic indicator. Regression models are used because unlike autoencoders used in the broader levels(furnaces/zones), regression can predict a target in output which is useful for diagnosing thermocouple performance. This is a supervised approach, unlike the models used at the furnace and zone levels.
- Create a Data Processing and Analysis Pipeline: To develop a pipeline capable of processing new or historical multivariate time-series data on a row-by-row basis, applying the level-based model structure that was built, and logging relevant outputs including reconstruction errors, predicted vs. actual TC values, and anomaly flags.
- Design an Interactive Visualization Dashboard: To implement a user interface that visualizes the system's outputs in near real-time, displaying anomaly statuses and error metrics hierarchically (furnace, then zone, then thermocouple), and allowing for the exploration of historical anomalies. The interactive dashboard, also has a dedicated page for summary of all the data points available to give an overview of the performance of the furnaces and how the anomaly scores has changed over time.

Given the lack of labeled anomaly data that would allow for quantitative testing, evaluating the system's effectiveness will need expert operational insights. Future validation will rely on feedback from the engineers within a live or simulated environment, as direct testing against known anomalies is not currently feasible.

This thesis is structured as follows: Chapter 2 reviews the relevant literature, focusing on unsupervised anomaly detection techniques, particularly autoencoders, and their application in industrial settings. It also covers regression techniques for sensor validation and the challenges involved in evaluating unsupervised systems. Chapter 3 outlines the methodology, describing the architecture and training process for both furnace-level and zone-level autoencoders, the development of thermocouple regression models, and the logic behind the level-based anomaly detection system. Chapter 4 details the implementation of the end-to-end system, including a comprehensive overview of the Outokumpu dataset (covering all six zones with their specific thermocouple configurations, fuel flows, and speed variables), the data processing pipeline for row-by-row analysis, and the design and features of the interactive dashboard for results visualization. Chapter 5 presents and discusses the results obtained from applying the system to operational data. Finally, Chapter 6 concludes the thesis by summarizing its contributions, acknowledging its limitations, particularly the difficulty of quantitative evaluation without ground truth labels, and suggesting directions for future work.

2 Background

This chapter introduces the essential background required to understand the methods, tools, and data used in this thesis and then delves into some essays and research related to this thesis. It begins by exploring the industrial context in which the anomaly detection system is deployed, focusing on the critical role of temperature monitoring in stainless steel production. Following this, it discusses key concepts from machine learning and deep learning, particularly those relevant to unsupervised anomaly detection in time-series sensor data.

2.1 Preliminaries

This section provides the necessary background to understand the methods and technologies used in this work. It introduces the industrial context, essential machine learning principles, and key concepts related to anomaly detection in time series sensor data.

2.1.1 Anomaly Detection

Anomaly detection plays a crucial role in various industrial applications by identifying deviations from expected operational patterns. In manufacturing and quality control, it's used with computer vision and sensor data to spot defects in products or packaging and detect equipment malfunctions, which leads to reduced downtime and improvement in product quality. For IT systems management, anomaly detection monitors server logs and system performance to predict potential failures and ensure smooth operations. Furthermore, in energy, transportation, and critical infrastructure sectors, it analyzes data from IoT and operational technology devices to predict equipment maintenance needs and identify irregularities in consumption or usage (IBM, 2020).

In this case, anomaly detection is used to identify unusual and unexpected behavior from the furnaces and zones based on the hidden patterns that exists in the time series data and is learned by the models during training stage. Detecting such anomalies early, helps with Predictive maintenance (identifying failing components before breakdowns). No prior anomaly detection system has been implemented specifically for these furnaces, making this work a novel contribution in this context.

Due to the lack of labeled data, supervised learning methods are often unsuitable, motivating the use of **unsupervised learning techniques**.

2.1.2 Unsupervised Learning

Unsupervised learning includes machine learning techniques that infer patterns from unlabeled data. Unlike supervised learning, where input-output pairs are known and models are trained to predict labels, unsupervised methods aim to find intrinsic structures or anomalies in the data without explicit guidance (IBM, 2021b).

In this thesis, unsupervised learning is the natural choice because the furnace data is unlabeled and also the normal process behavior dominates the dataset. Therefore, unsupervised learning allows the model to learn what is “normal,” so that anything significantly different can be flagged for further inspection. For these flagged points to reliably indicate real anomalies and potential malfunctions, the system requires thorough implementation and testing. This iterative approach is crucial for measuring accuracy and continually refining the models over time.

2.1.3 Artificial Neural Networks (ANNs) & Deep Learning

Artificial Neural Networks (ANNs) are computational models inspired by biological neural systems. They consist of layers of interconnected neurons, where each neuron calculates a weighted sum of its inputs ($\sum w_i x_i$ + bias) before passing it through a non-linear activation function to determine its output. Training these networks involves minimizing an error function, like the Mean Squared Error (MSE), using backpropagation, an algorithm that updates the network's weights based on the gradient of the loss function. The key difference between a basic network and a deep one is its depth. A "shallow" or basic network might only have three layers total (an input, a single hidden, and an output layer). In contrast, Deep Learning refers to ANNs with multiple hidden layers, allowing the network to capture more complex, hierarchical representations in the data. The general structure of deep neural network can be seen in Figure 1. This added depth has proven particularly effective for tasks involving high-dimensional or unstructured data, making it well suited for anomaly detection in sensor systems (IBM, 2021a).

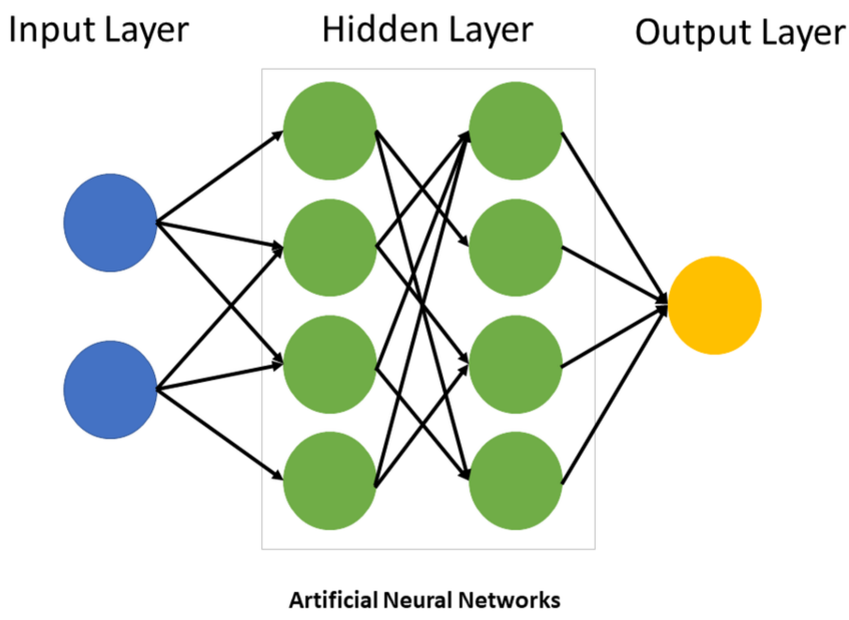

Figure 1: Structure of a deep neural network

2.1.4 Autoencoders (AEs)

An autoencoder represents a specific type of artificial neural network architecture crafted for the purpose of efficiently compressing input data into a more condensed form, known as its latent space representation. Subsequently, the network is trained to reconstruct the original input data from this compressed representation. This process leverages unsupervised machine learning, and enables the autoencoder to discover efficient data encodings without requiring any labels. The primary aim is to learn a representation (encoding) for a set of data, typically for dimensionality reduction, by training the network to ignore signal "noise". As a result, the latent space representation captures only the essential information from the original input (Provotar et al., 2019).

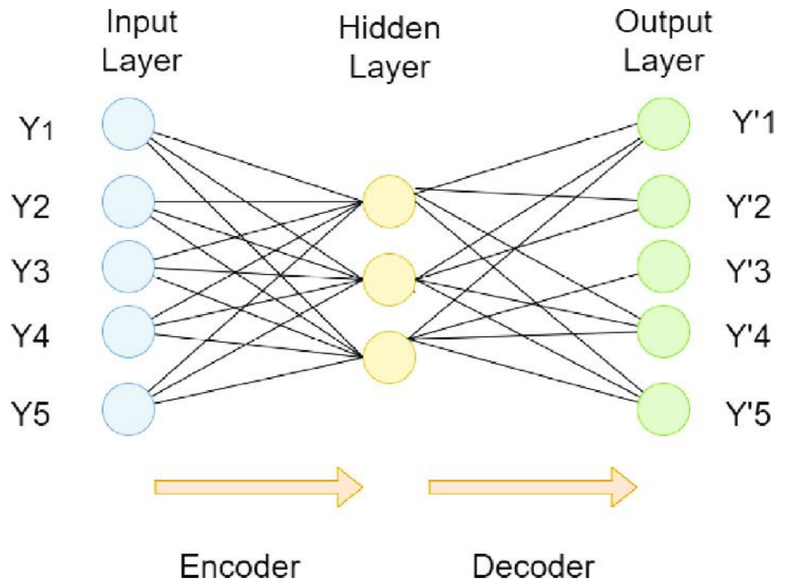

Figure 2: Network Architecture for an Ordinary Autoencoder

As can be seen from Figure 2, the operational mechanism of an autoencoder involves two main modules: an encoder and a decoder. The encoder learns the underlying features of a process, typically in a reduced dimension, and creates a compressed representation. This compressed form, which can be thought of as a "bottleneck," serves as the output of the encoder and the input to the decoder. The decoder then reconstructs the original data from these underlying features. This reconstructed output is compared to the original input to assess the autoencoder's correctness; the difference between them is called the reconstruction error (Provotar et al., 2019).

To validate the performance of the proposed autoencoder and since it is an unsupervised approach, a benchmark comparison was done against a standard machine learning algorithm, Isolation Forest. This method was chosen for two primary reasons: its computational efficiency and its fundamentally different, isolation-based approach to anomaly detection, which provides a strong methodological contrast to the reconstruction-based autoencoder (Wu et al. ,2022). The correlation between the anomaly scores of the two models was used to assess the degree to which they identify the same events. The ultimate validation and refinement of such a system will inevitably depend on its deployment and the incorporation of feedback from domain experts.

2.1.5 Isolation Forest

Isolation Forest is an unsupervised machine learning algorithm highly effective for anomaly detection. Isolation Forest works by directly isolating anomalies. It does this by randomly selecting features and splitting data points until they are separated. Anomalies, being rare and different, typically require fewer splits to be isolated in these random partitions (GeeksforGeeks, 2024).

The algorithm builds an ensemble of isolation trees. Each tree is built by recursively partitioning the data points through random feature selection and random split points until individual data points are isolated. Each data point is assigned an anomaly score based on the average number of splits it takes to isolate it across all trees; the fewer the splits, the higher the anomaly score, which shows a higher likelihood of being an outlier. This approach makes Isolation Forest computationally efficient and robust, particularly in high-dimensional datasets (GeeksforGeeks, 2024).

2.1.6 Regression Analysis

Regression is a supervised learning technique used to predict a target variable based on input features. In this project, regression models, specifically ridge regression models, are used as part of a diagnostic layer: to estimate expected thermocouple values given correlated sensor inputs in the same zone.

By comparing these predicted values with actual readings, it becomes possible to Identify which thermocouple is likely to be faulty.

Ridge regression is a statistical technique employed in machine learning to prevent overfitting in models, particularly linear regression models. Also known as L2 regularization, it's especially useful when dealing with multicollinearity, a situation where independent variables in a dataset are highly correlated. This method addresses overfitting by introducing a "penalty term" into the traditional regression equation. This penalty works to reduce, or "shrink," the values of high regression coefficients, making the model less sensitive to minor fluctuations in the training data (IBM, 2023).

2.2 Research background

Detecting anomalies in time series presents a significant challenge in data science. As previously noted, anomaly detection techniques can be categorized based on the availability of data labels. With a labeled dataset, accurate models can be developed using classification methods. However, the majority of datasets lack labels, and the labeling process necessitates the involvement of domain experts. Furthermore, manually labeling extensive datasets is a time-intensive and costly endeavor.

Although established statistical methods provide fundamental tools, the complexities arising from high-dimensional sensor relationships and the common lack of labeled anomaly data in practical applications demand more flexible and data-centric solutions, largely using the techniques from the fields of unsupervised machine learning and deep learning. This section offers a review of some existing research relevant to unsupervised anomaly detection (Wu et al. ,2022).

Reconstruction-based models form a significant category within unsupervised anomaly detection methods, with Autoencoders (AEs) and their variations having substantial interest. The core idea is to train a model, commonly a neural network, to faithfully reconstruct normal operational data. Anomalous data points, being deviations from these learned normal behaviors, are anticipated to produce significant reconstruction errors. For instance, Amarbayasgalan et al. (2020) in their RE-ADTS system implemented a deep autoencoder on time-series subsequences, employing the reconstruction error as an indicator of anomalies, they used this strategy because traditional anomaly detection models often lack the capacity to effectively capture complex temporal dependencies and non-linear patterns inherent in time-series data, particularly in unsupervised settings where labeled anomaly data is not available.

To address the inherent multivariate nature and inter-sensor dependencies in industrial environments, more advanced architectures have emerged. The MSCRED (Zhang et al., 2019) and MCRAAD (Zhao et al., 2021) frameworks represent sophisticated direction in reconstruction-based methods. These models initially transform multivariate time-series data into "Signature" or "Feature" matrices that represent inter-sensor correlations within defined time windows, the main reason for doing so, is to consider the temporal dependencies between variables that is in the nature of time series data. Subsequently, a recurrent autoencoder architecture reconstructs these correlation matrices. A recurrent autoencoder is an autoencoder where the "recurrent" part specifically means it uses recurrent neural networks (RNNs), which are specially designed to process sequential data by having internal memory that allows them to consider past information in a sequence when processing current inputs. A significant reconstruction error in these learned relationships can signal an anomaly, and notably, the residual analysis in MSCRED offers a means to identify which sensor's correlations are most impacted. These approaches underscore the significance of considering the temporal relationship in the time series data, a concept relevant to understanding furnace zone interactions in this project.

Prediction-based anomaly detection offers an alternative, where models forecast future sensor values, and significant deviations between predictions and actual observations signal anomalies. Munir et al. (2019a) in DeepAnT utilized a 1D Convolutional Neural Network (CNN) for this purpose, directly using the forecast error as an anomaly score. 1D Convolutional Neural Network is a type of neural network specifically designed for processing sequential data, like time series. It uses one-dimensional convolutional filters to extract features and patterns from the input sequence, enabling it to learn and predict future values within the time series. Wu et al. (2022) demonstrated the feasibility and effectiveness of prediction-based anomaly detection in time series data, particularly in scenarios with complex multi-seasonality and without labeled data. Their unsupervised approach leverages a forecasting model to anticipate normal patterns and identifies anomalies as significant deviations from these predictions. For thermocouple-level diagnostics, as pursued in this thesis, a direct regression approach for predicting individual thermocouple readings based on their local sensor context (other TCs, fuel flows, etc.) aligns with this predictive philosophy but targets a more granular level of detail than system-wide forecasting. The literature supports the use of various regression techniques, including ridge regression, for modeling sensor behavior and identifying deviations when actual readings diverge from predictions derived from correlated variables.

While many advanced models focus on detecting anomalies at a system or subsystem level, in this thesis, there is a recognized need for approaches that can provide more localized diagnostics. The level-based approach adopted in this thesis, using autoencoders at the furnace and zone levels for broader anomaly detection, and then employing specific regression models for individual thermocouple validation, propose a new, initial system for comprehensive anomaly detection. The aim is to integrate these unsupervised autoencoder techniques for broader monitoring with targeted regression models for sensor-level diagnostics within a structured hierarchy, specifically designed for the complexities of industrial furnace operations and the critical task of identifying thermocouple-specific anomalies without pre-existing labels. The evaluation, therefore, focuses on the practical utility of this combined hierarchical system in an industrial setting where expert validation is key, rather than direct comparison on benchmark datasets with predefined anomalies.

2.3 Why is it important to understand these research papers?

To develop a methodology for this thesis, it was essential to examine the existing research papers in unsupervised anomaly detection. This review provided a foundation in established machine-learning-based concepts that can be implemented as an initial system for these furnaces. The insights gained from previous studies, especially those addressing the complexities of unlabeled, high-dimensional time-series data, have directly shaped the design decisions for this project.

One of the most significant takeaways from the literature is the effectiveness of autoencoders for unsupervised anomaly detection, which is particularly relevant to industrial environments where labeled anomaly data is often scarce or in this case, non-existent. Their inherent ability to learn compact representations of normal operational behavior, and then flag deviations through increased reconstruction error, makes them an ideal choice for identifying abnormalities in the furnace sensor data explored in this thesis (Amarbayasgalan et al., 2020; Wu et al., 2022). Consequently, this research uses a level-based autoencoder architecture. This involves a furnace-level autoencoder to monitor overall system behavior, complemented by zone-level autoencoders that provide deeper detection by isolating anomalies within specific furnace zones.

While autoencoders are good at flagging the presence of an anomaly, interpreting the root cause from a single reconstruction error score can be challenging. In industrial settings, such actionable diagnostic information is as important as the initial detection. To address this, this thesis tries to enhance interpretability in two ways. First, for any anomaly detected by an autoencoder, the system calculates the feature-wise reconstruction error. By analyzing the individual squared error for each input feature, we can identify which specific sensors or process variables are contributing most to the anomaly score. This provides an immediate, ranked list of potential culprits for an operator to investigate. Second, to provide even more precise diagnostics specifically for the critical thermocouple (TC) sensors, this thesis uses dedicated regression models. Drawing inspiration from prediction-based methodologies that were discussed, these models are designed to predict expected temperature values based on inputs from related sensors in their zone (such as other thermocouples and flow values inside the same zone). By comparing these predicted values against actual readings, a residual error is computed, which gives us a localized explanation for anomalies. This approach enables the system to move beyond zone-level identification and provide thermocouple-specific error signals.

Furthermore, a recurring theme in many reviewed studies, including advanced architectures like MSCRED (Zhang et al., 2019) and MCRAAD (Zhao et al., 2021), is the critical role of temporal dependencies in time-series data. These works clearly demonstrate that a model's performance is significantly enhanced when it can consider historical context. Drawing from this key insight, this thesis incorporates temporal information in a direct and efficient manner by augmenting the feature set with lagged variables. By including the state of each sensor from the previous timestamp (t-1), all models in the system are given a basic "memory," allowing them to learn from the most recent dynamics of the process. The dramatic improvement in the performance of the diagnostic regression models after adding these lags serves as direct validation of this approach. While more complex temporal architectures like the ones mentioned, could potentially capture longer-term dependencies, a deliberate choice was made to start with this simpler, more interpretable, and computationally effective method as a robust foundational step. In future iterations, more advanced time-aware models can be implemented to capture subtle patterns like gradual sensor drift.

3 Method

This thesis adopts an experimental approach to explore unsupervised anomaly detection in multivariate industrial time-series data. The core objective is to detect anomalous patterns within sensor readings from a steel furnace without relying on labeled data. This approach is suitable due to the nature of the dataset, which lacks annotated anomalies, and the industrial context, where manual labeling is expensive and often unavailable.

The research involves the development, evaluation, and refinement of a level-based detection system. The goal is to produce a functional and scalable system capable of identifying anomalies at different levels of granularity, from overall furnace behavior to specific thermocouple sensors, this will help the engineers to have a better idea on where to focus on in case of abnormal behavior from the system.

3.1 Method Selection and Justification

Given the lack of labeled data, this study employs unsupervised learning techniques. Specifically, autoencoders were chosen for their ability to learn compressed representations of normal operational data and signal deviations via reconstruction errors and regression models (e.g., Ridge Regression) were used to predict thermocouple readings based on correlated sensor inputs to identify localized anomalies through residual analysis.

Reconstruction-based models (e.g., autoencoders) are well-suited to learning normal behavior, while prediction-based methods provide explainable errors at the sensor level. The combination enables both broad anomaly detection and detailed diagnostics.

To validate the proposed autoencoder, and because it's an unsupervised approach, a benchmark comparison was conducted against Isolation Forest, a standard machine learning algorithm. The correlation between the anomaly scores of the two models was then used to assess their agreement on identified events.

Alternative approaches such as supervised classification or rule-based systems were considered unsuitable due to the absence of labels and the complexity of manually encoding operational rules for this high-dimensional data.

3.2 Level-Based Framework Design

The detection system is structured hierarchically:

- **Furnace-level autoencoder**: Trained on all sensor features inside that furnace to capture general operational behavior.
- **Zone-level autoencoders**: Focused on each furnace zone to isolate anomalies more precisely when detected at the furnace level. The models are trained on all the data withing that zone.
- **Thermocouple regression models**: Applied only when a zone-level anomaly is detected, these models predict each thermocouple's value using its surrounding sensor context (thermocouples, flow transmitters and other variables inside that zone this thermocouple resides in). The operators can see the amount predicted and the true value captured by the sensor.

This structure allows the system to perform progressive refinement, moving from global anomaly detection to zone-specific and eventually sensor-specific diagnostics.

3.3 Data Collection and Preprocessing

The dataset was provided from an industrial steel furnace by *Outokumpu* and contains zone-wise sensor readings, including thermocouples (temperature sensors) and flow transmitters, collected at one-minute time intervals.

Preprocessing included:

- Normalizing sensor values using standard scalers. Scaling used is StandardScaler from sklearn.preprocessing.
- For each model, the variables needed for training that model (for a specific zone, furnace, or thermocouple) were chosen from the dataset.

4 Implementation and results

4.1 Dataset

The dataset used for this project was provided by Outokumpu, containing detailed sensor data from two industrial furnaces (which are attached together, each composed of multiple zones. Each zone records several process variables, including fuel flow rates, steel temperature, and multiple thermocouple readings. The data was segmented as follows:

- Furnace 1: Zone 1 and Zone 2
- Furnace 2: Zone 3, Zone 4, Zone 5, Zone 6

Each zone includes:

- Fuel Flow: Measurements of gas or oil flow to burners.
- Thermocouples (TC): Multiple temperature sensors per zone.
- Steel Temperature and Strip Speed: Indicators of overall furnace performance.

All the variables are numerical expect DateTime.

The dataset comprises thermocouple readings collected over five months, from December 2024 to April 2025. Each month's worth of data is stored separately, with the individual monthly files containing between 33,121 and 44,641 rows. These readings are recorded at one-minute intervals, providing a detailed time-series record of the thermocouple measurements. All five monthly datasets are then combined to form a comprehensive dataset, totalling 207,305 rows.

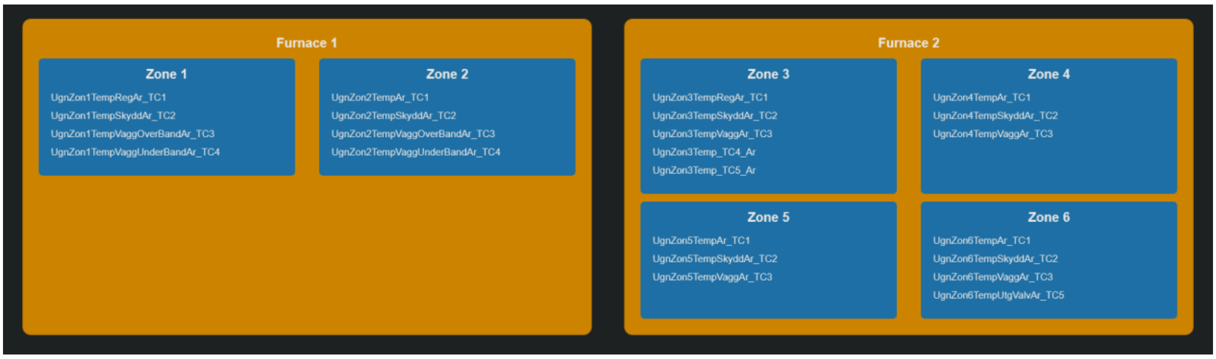

Figure 3: zones of each furnace and thermocouples inside each zone

As shown in Figure 3, The dataset is systematically divided into six distinct operational zones across two furnaces, with two zones for furnace 1 and four zones for furnace 2. Each zone's data focuses on relevant parameters including average steel temperature, various fuel flow measurements, and multiple strategically placed thermocouples. A consistent timestamp (DateTime) and a global strip speed/control output (LineControlHastSverk4_1Act) are included in all zone-specific datasets.

Furnace 1 Zones

Zone 1: The dataset for Zone 1 consists of ten variables. It includes three fuel flow measurements (upper, lower, and total), four temperature readings (regulation, safety, upper wall, and lower wall), the average steel temperature, line speed, and a timestamp.

Zone 2: Zone 2 has ten variables. These comprise three fuel flow inputs (upper, lower, and total), four temperature sensors (regulation, safety, upper wall, and lower wall), along with the average steel temperature, line speed, and a timestamp.

Furnace 2 Zones

Zone 3: Zone 3 contains eleven variables. This includes three fuel flow measurements (upper, lower, and total) and five temperature sensors: one for regulation, one for safety, a wall sensor, and two additional thermocouples. The average steel temperature, line speed, and a timestamp.

Zone 4: Zone 4 has ten variables. The data captures three fuel flow measurements (upper, lower, and total) and four temperature readings from regulation, safety, and two separate wall sensors. The average steel temperature, line speed, and a timestamp.

Zone 5: Zone 5 has seven variables. It features a single total fuel flow measurement and three temperature sensors for regulation, safety, and wall temperature. The average steel temperature, line speed, and a timestamp.

Zone 6: Zone 6 has eight variables. Data includes a single total fuel flow reading and four temperature measurements from regulation, safety, wall, and furnace outlet sensors. The average steel temperature, line speed, and a timestamp.

4.2 Feature Selection

To ensure each model operates on relevant and localized data, a consistent feature engineering step was applied first: for every sensor reading, its value from the previous timestamp (t-1) was added as a new "lagged" feature. This augmented dataset, containing both current and recent historical information, was then used for training.

Furnace-Level Autoencoders were trained on all available columns from all zones' data within each furnace.

Zone-Level Autoencoders were trained only on that specific zone's data — fuel flow, average steel temperature, strip speed, and all thermocouple values within the zone.

Thermocouple Regression Models were trained per thermocouple using all other zone-specific variables (except the TC being predicted).

This structure ensures that each model has the most contextually relevant data and avoids overfitting to unrelated zones.

4.3 Model Development

4.3.1 Autoencoders (AEs)

Each zone was modeled using an unsupervised autoencoder trained to reconstruct its input features, so it learns a compressed representation of normal operational behavior. The core idea behind this method is that the autoencoder becomes proficient at reproducing inputs it has seen frequently, which are patterns from normal operation, but performs poorly when encountering unfamiliar or abnormal data. This property enables the model to flag potential anomalies when the reconstruction error (mean squared error between the input and output) exceeds a certain threshold.

The data splitting is handled by train_test_split with a key configuration: shuffle=False. This ensures that the dataset is divided chronologically, with earlier data points forming the training set and later points reserved for testing, preventing any look-ahead bias. The test size is 20%. Before this split, the DateTime column and any rows with missing values are removed. Following the split, StandardScaler is employed for normalization: it's fit exclusively on the training data to learn the scaling parameters, and then both the training and test sets are transformed using these parameters, preventing data leakage from the test set during the scaling process.

The autoencoder architecture was designed to be adaptable across different zones with varying numbers of input features. It begins with an Input layer, which the shape of it dynamically matches input_dim, representing the number of features in the pre-processed data for each zone. This input layer is followed by two Dense (fully connected) hidden layers that makes the encoder. Their neuron counts are calculated relative to the input_dim: the first encoder layer has (input_dim * 0.75) neurons, and the second, acting as the bottleneck, has (input_dim * 0.5) neurons. Both encoder layers utilize the ReLU (Rectified Linear Unit) activation function.

Following the bottleneck, the decoder part mirrors the encoder's structure. It consists of two Dense layers: the first decoder layer has (input_dim * 0.75) neurons, also with ReLU activation, aiming to progressively expand the compressed representation. The final Dense output layer has input_dim neurons, matching the original input dimensionality, and employs a linear activation function. A linear activation is essential here because the. Future fine-tuning of layer widths, depths, and activation functions will be crucial, and will be guided by the performance metrics observed during deployment and the specific characteristics of anomalous patterns encountered.

After training, the autoencoder was used to reconstruct the inputs from the test set, and the reconstruction errors were computed for each test sample. These errors were then used to define an anomaly detection threshold. Specifically, the threshold was set to the 95th percentile of the validation reconstruction errors. This means that, under normal operating conditions, 95% of the test samples should fall below this threshold. Any future input whose reconstruction error exceeds this threshold is flagged as a potential anomaly, which indicates behavior that deviates significantly from the learned patterns of normal operation. This percentile is a placeholder and it can easily be configured over time depending on the needs and observations of anomalies in operation, lowering it would flag more points as anomalies and increasing it will reduce the number of flagged anomalies.

4.3.2 Isolation forest

Data preprocessing for the Isolation Forest model mirrors the approach used for the autoencoder, in terms of splitting and scaling data.
Unlike reconstruction-based methods such as autoencoders, Isolation Forest works by directly isolating anomalies rather than profiling normal data. It does this by randomly selecting a feature and then a random split value for that feature, recursively partitioning the data until individual instances are isolated. Anomalies, are typically isolated closer to the root of the tree, requiring fewer splits.

After training, anomaly scores are calculated for the test set. By default, the model returns higher values for normal instances and lower (more negative) values for anomalous ones. To align with a more intuitive understanding where higher values indicate greater anomaly, these scores are inverted.

Finally, an anomaly threshold is determined by taking the 95th percentile of these inverted anomaly scores from the test set. The threshold can be configured like the threshold from autoencoders.

4.3.3 Thermocouple-Specific Regression Models

To achieve anomaly localization within the furnace system, thermocouple-specific regression models were trained for individual thermocouples inside zones. Each thermocouple was modeled separately using ridge regression. The purpose was to predict each thermocouple’s temperature reading based on all other available features from the same zone, excluding the target thermocouple itself. The same data preprocessing as the previous models was applied here

Each model was evaluated using Root Mean Squared Error (RMSE) on test sets to assess performance. Once trained, both the model and its corresponding scaler were saved as .pkl files, structured per thermocouple and zone, for later deployment in the anomaly diagnosis pipeline.

Figure 4: Ridge regression results for furnace 1's thermocouples

Figure 5: Ridge regression results for furnace 2's thermocouples

As shown in Figure 4 and 5, the results are very good for all the thermocouples. The RMSE values are the degree in Celsius, which considering the average values of each thermocouple, is a very small value, making the models very accurate in their prediction. The results were even more precise than the results gained from ridge models trained on data without adding "lag". Table 1 in appendix contains the results for the previous models trained on data without lag. This signifies the importance of having a memory for this time series data when it comes to predicting thermocouple values.

4.4 Validation and Benchmark

To validate the performance of the proposed autoencoder-based system in the absence of labeled data, a benchmark comparison was conducted against a standard, established unsupervised learning algorithm: Isolation Forest.

For each of the eight autoencoder models (two furnace-level and six zone-level), a corresponding Isolation Forest model was trained on the exact same training data with lag. To create a fair comparison of their outputs, the anomaly scores from both models on the test set were evaluated. The autoencoder's score was its reconstruction error, while the Isolation Forest's score was derived from its score_samples method (inverted so that a higher value indicates a higher anomaly).

since the two models operate on different internal scales, their raw scores were normalized to a common [0, 1] range using Min-Max scaling. This allows for a direct visual comparison of their relative behavior over time. The primary metric for quantitative comparison was the Pearson correlation coefficient, calculated between the two normalized score series. This metric measures the degree to which the two models' anomaly scores rise and fall together in response to events in the data.

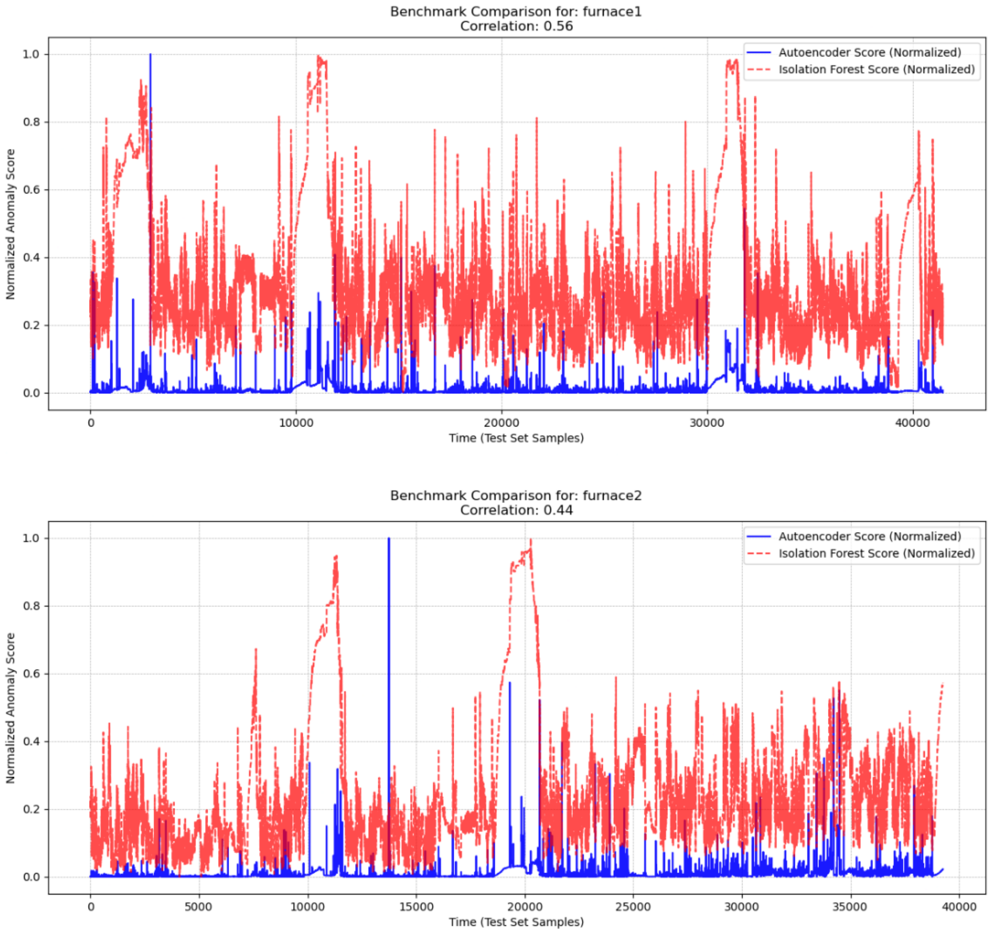

Figure 6: Benchmark comparison for both furnaces

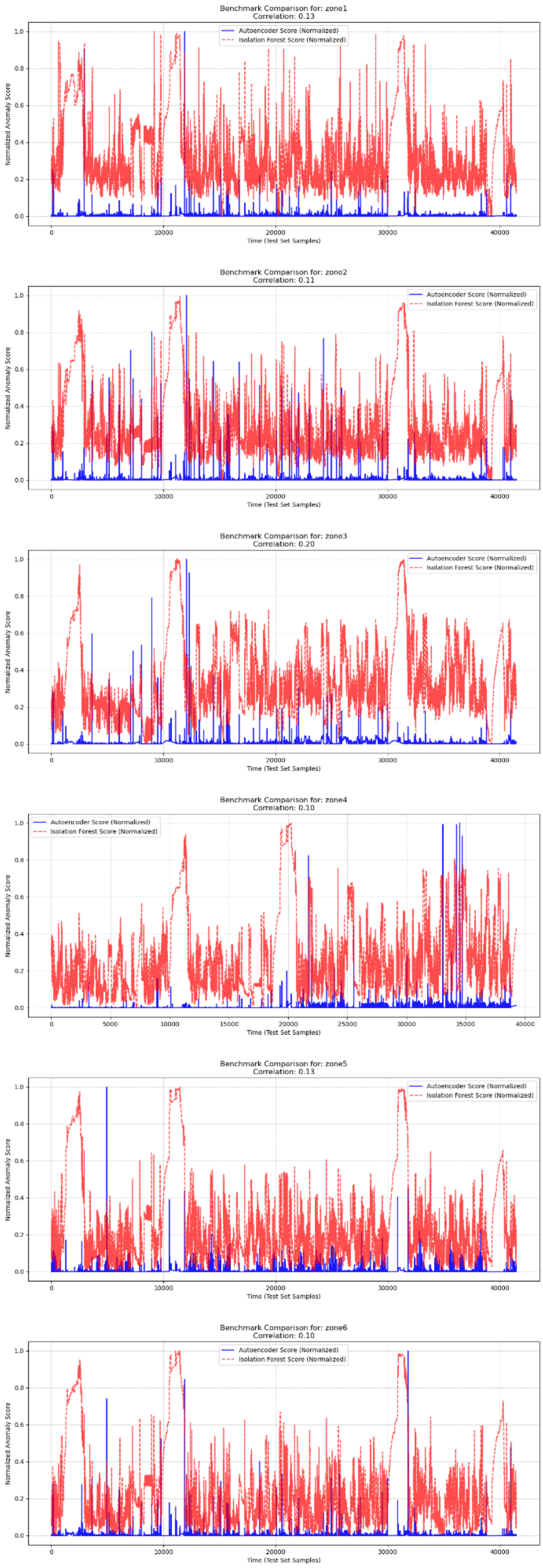

Figure 7:Benchmark comparison for zones

The comparison revealed two distinct patterns. At the furnace level, shown in Figure 6, the models showed a moderate positive correlation (0.56 and 0.44). This indicates that for significant, system-wide events that affect the entire furnace, both the autoencoder and the Isolation Forest tend to agree that an anomaly is occurring. This provides a strong validation that the top-level detection capability of the proposed system is reasonable and aligned with standard methods.

At the zone level, shown in Figure 7, a weak but consistently positive correlation was observed (ranging from 0.10 to 0.20). This lower correlation does not indicate a failure of either model. Instead, it suggests that the models have different specializations and are sensitive to different types of anomalies.

The Autoencoder, having learned the complex, non-linear inter-relationships between all features, is considered to excel at detecting contextual anomalies, where the system's state violates a learned operational rule. The Isolation Forest, with its random splitting mechanism, is naturally more adept at identifying point anomalies, where a single sensor reading is an extreme value, regardless of context.

4.5 Anomaly Detection Pipeline

The deployment of this anomaly detection system in a live industrial environment can be realized in various ways, depending on the specific infrastructure and operational requirements. The following description details a prototype implementation, developed as a proof-of-concept to demonstrate the end-to-end functionality of the proposed methodology.

A Streamlit-based monitoring interface, depicted in Figure 8, was developed to perform real-time anomaly detection by processing the dataset row by row. Streamlit is an open-source Python framework that allows us to quickly build and deploy interactive web applications with minimal code; the concepts and visualization pipeline describe here can be easily transferred to other visualization tools like Power bi. The detection pipeline follows a hierarchical structure to ensure computational efficiency. At the first level, the system performs a furnace-level check, which is executed for every incoming data row. Autoencoders trained for **Furnace 1** and **Furnace 2** are used to compute reconstruction errors, and if the error for a furnace exceeds its predefined threshold, that furnace is flagged as anomalous.

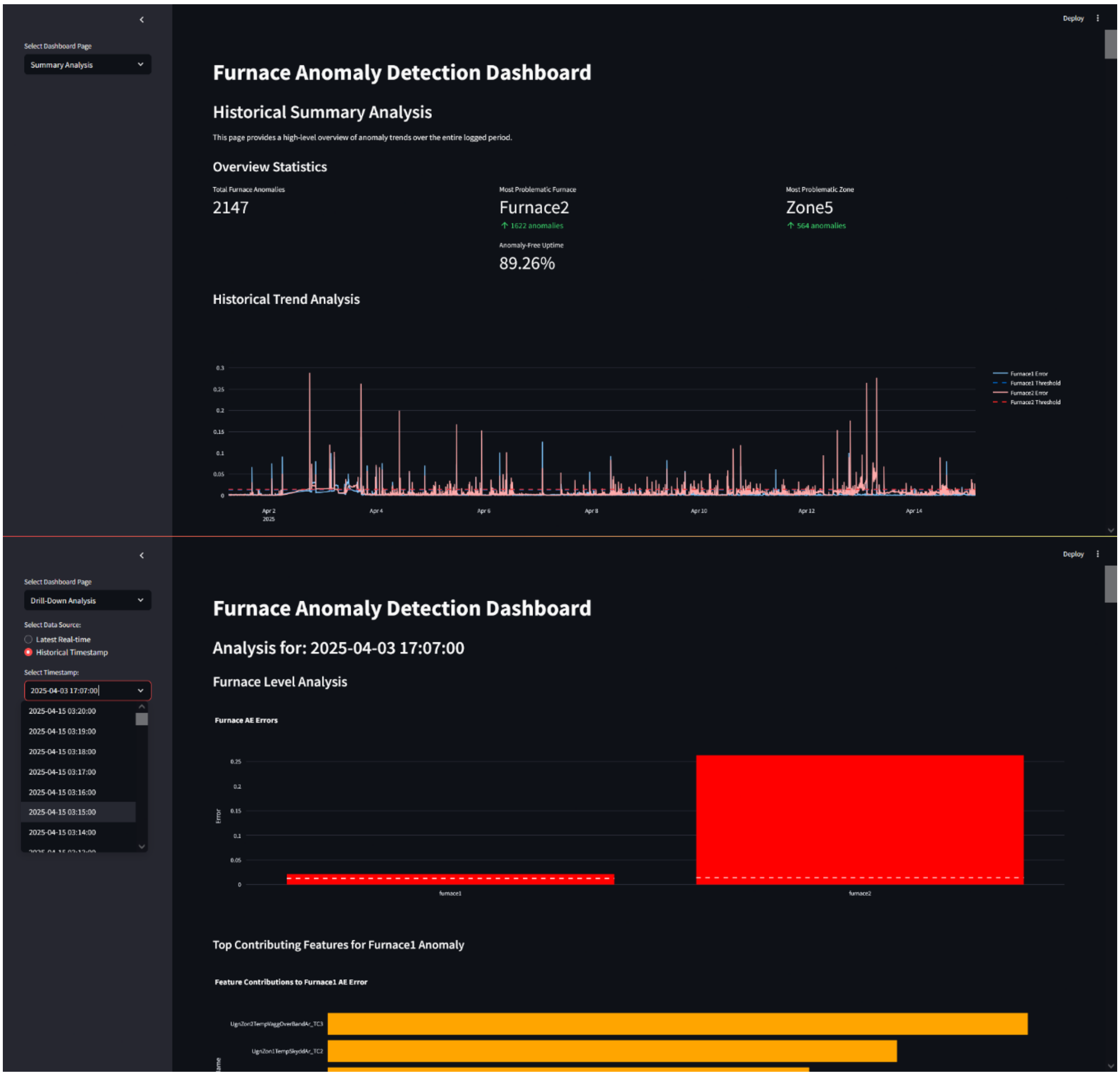

Figure 8: Overview of the Anomaly Detection Dashboard, Displaying both the detailed row by row analysis and the summary of all datapoints pages

If a furnace is flagged, the system conditionally performs a **zone-level check**. All zone-specific autoencoders within the flagged furnace are run, and any zone whose reconstruction error surpasses its threshold is added to a list of anomalous zones. To further pinpoint the issue, a thermocouple regression step is selectively executed. Among the flagged zones, the one with the highest autoencoder error is selected for deeper inspection. Regression models trained specifically for thermocouples in that zone are used to estimate expected sensor readings. The residuals (i.e., prediction errors) are then calculated and logged for interpretability by the operators.

This layered strategy, starting at the furnace level, going down to zones, and finally inspecting thermocouple readings, ensures that only relevant computations are performed, which reduces resource usage while still offering actionable insights into the nature and location of anomalies.

To support quick and intuitive analysis, the Streamlit app dynamically generates layered visualizations. At the furnace level, bar charts display the reconstruction errors against their respective thresholds, clearly indicating when anomalies occur. If a furnace is flagged, a zone-level view becomes available, showing the reconstruction errors of each zone inside that furnace to highlight which areas are contributing to the anomaly. If thermocouple-level regressions are triggered, the UI presents thermocouple-level visualizations comparing predicted and actual values. This multi-tiered visualization strategy allows engineers and operators to trace anomalies from high-level system behavior down to specific sensor discrepancies with ease.

To enhance the system's interpretability and provide actionable insights for operators, a feature contribution analysis was integrated into the diagnostic pipeline. When an autoencoder at either the furnace or zone level flags an anomaly, the system does not just report the total reconstruction error. It also calculates the individual squared error for each input feature. This process deconstructs the total error, revealing precisely which sensor signals the model found most difficult to reconstruct. In the Streamlit dashboard, these contributions are visualized as bar chart, as depicted in Figure 9, presenting a ranked list of the top five most influential features. This immediately directs the operator's attention to the specific sensors or process variables that are the likely root cause of the anomaly, transforming a simple alert into a powerful, targeted diagnostic tool.

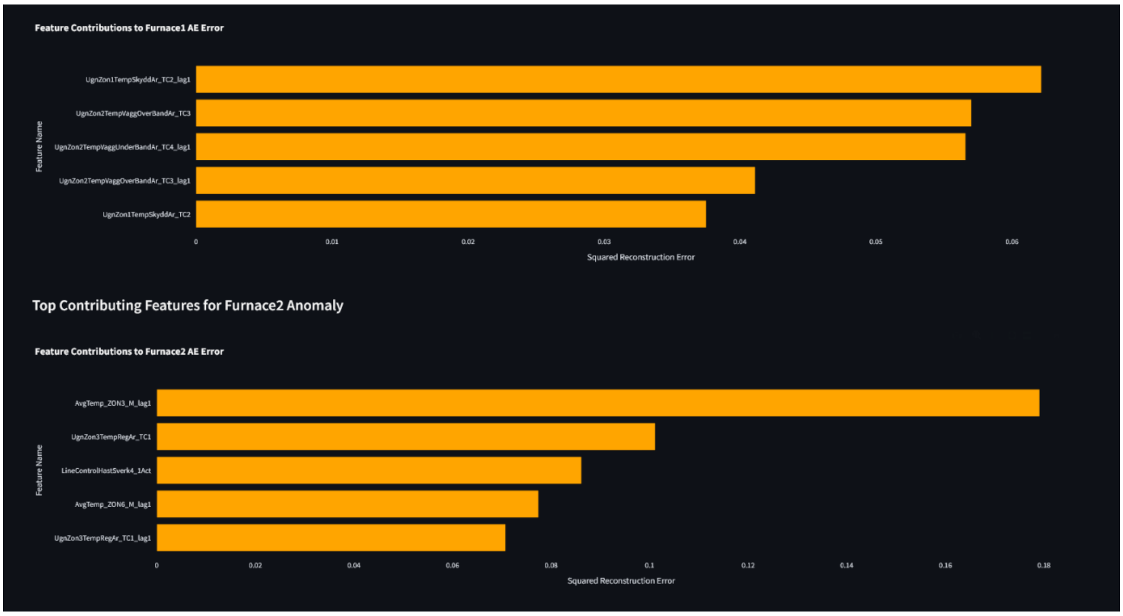

Figure 9: Feature contribution bar charts

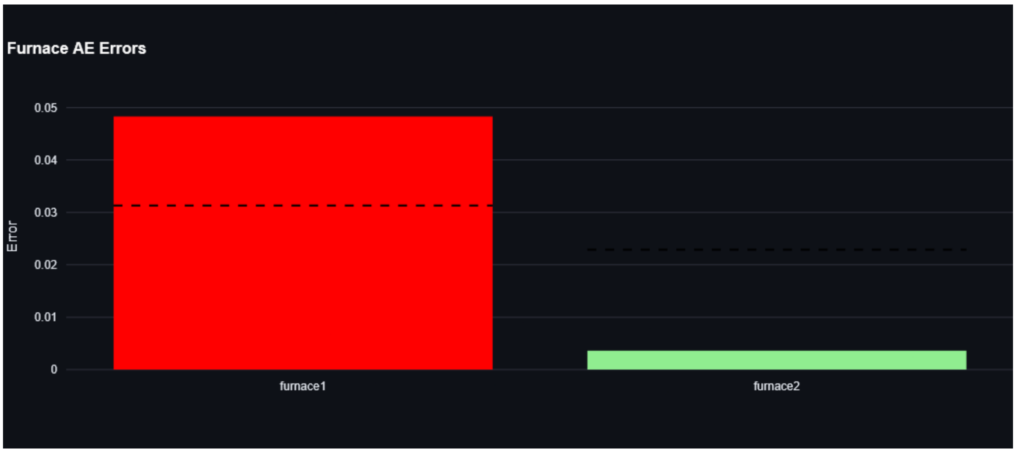

Figure 10: Example of Furnace-Level Anomaly Detection – Bar Chart of Reconstruction Errors for Furnace 1 (Anomalous) and Furnace 2.

In an example scenario, furnace 1 shows an anomaly, because the reconstruction error exceeds the threshold of that model (Figure 10). Therefore, the models for zones inside furnace 1, that are zones 1 and 2 are used (Figure 11). We can also see the most influential features in the error of this zone (Figure 12). As zone 2 is the anomalous zone, the regression models for all of its Thermocouples are used so we can observe the prediction against its true value (Figure 13)

Figure 11: Zone-Level Anomaly Drill-Down – Bar Chart of Reconstruction Errors for Zones within Anomalous Furnace 1.

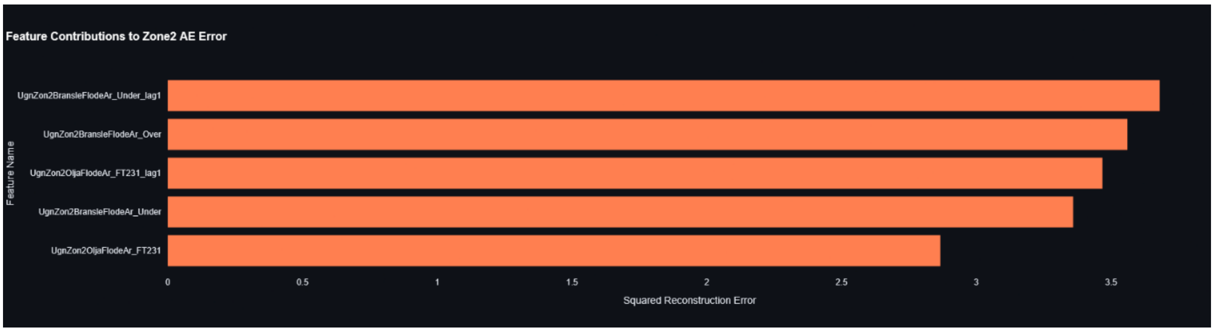

Figure 12: Feature contribution to zone2 AE error

Figure 13: Thermocouple-Level Diagnostics – Comparison of Predicted vs. Actual Values of degree Celsius for Thermocouples in Anomalous Zone 2 (from Furnace 1)

Beyond real-time monitoring, a dedicated "Summary Analysis" page was developed within the dashboard to provide a high-level, historical overview of the system's performance. This page analyzes the entire anomaly_log.csv to reveal long-term trends and patterns. So this can be the data for hours, days or weeks. It features several key components:

- Time series interactive plots for each furnace and zone, displaying the reconstruction error against its threshold over time, with anomalous points highlighted to visualize the frequency and magnitude of past events (Figure 14).
- A feature importance summary, which aggregates the contribution scores across all anomalies to rank the features that most consistently cause high reconstruction errors for a given furnace or zone (Figure 14).
- An anomaly frequency chart, which provides a simple count of total anomalies per zone, helps identifying the most problematic areas (Figure 15).
- A table of the most severe events lists the top historical anomalies sorted by their reconstruction error, allowing engineers to quickly review and analyze the most significant past incidents (Figure 16,17).

Figure 14: Time series anomaly analysis for furnaces and zones with the sum of their most influential factors over that time

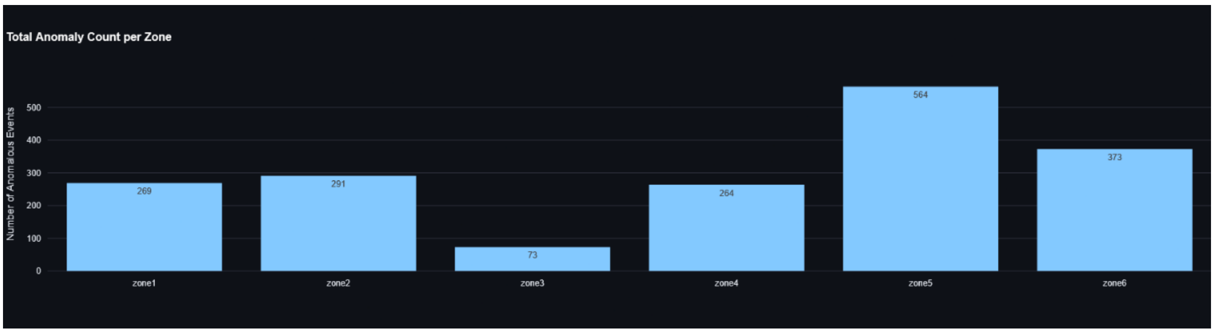

Figure 15: Total anomaly count per zone

Most Severe Anomalous Events by Furnace

	timestamp	max_error	furnace_furnace1_is_anomalous	furnace_furnace2_is_anomalous
2171	2025-04-02 18:12:00	0.289	☑	☑
17352	2025-04-13 07:13:00	0.2769	☑	☑
17084	2025-04-13 02:45:00	0.2654	☑	☑
3546	2025-04-03 17:07:00	0.2629	☑	☑
4573	2025-04-04 10:14:00	0.1998	☐	☑
16644	2025-04-12 19:25:00	0.1766	☐	☑
6104	2025-04-05 11:45:00	0.1677	☑	☑
16299	2025-04-12 13:40:00	0.1541	☑	☑
6774	2025-04-05 22:55:00	0.1531	☑	☑
3548	2025-04-03 17:09:00	0.1377	☑	☐

Figure 16: Most severe anomalous events by furnace

Most Severe Anomalous Events by Zone

	timestamp	max_error	zone
40	2025-04-03 17:07:00	2.1603	Zone5
41	2025-04-02 18:12:00	1.9263	Zone5
10	2025-04-03 19:52:00	1.2232	Zone2
50	2025-04-03 17:07:00	1.151	Zone6
42	2025-04-05 09:16:00	1.0045	Zone5
11	2025-04-01 16:26:00	0.9832	Zone2
0	2025-04-03 17:08:00	0.9828	Zone1
12	2025-04-12 07:26:00	0.9379	Zone2
30	2025-04-13 07:13:00	0.9263	Zone4
31	2025-04-12 07:26:00	0.92	Zone4

Figure 17:Most severe anomalous events by zone

5 Discussion

5.1 Methods, implementation and results

The implemented system, featuring a hierarchical anomaly detection framework using furnace and zone level autoencoders followed by thermocouple-specific regression models, shows potential for practical deployment in industrial environments. Its design aligns with a central goal of detecting anomalies without relying on labeled data, while still enabling diagnostic insights. The results demonstrate that this architecture can flag anomalies at multiple levels and provide sensor-specific diagnostic cues, which could serve as a valuable tool for process engineers and operators.

However, several important caveats must be addressed regarding the interpretation and generalizability of these results.

5.1.1 Expert Validation

While the system outputs look reasonable and the logic behind model predictions is consistent with expected anomaly behavior, the current results are not validated by domain experts. As such, the practical reliability of the model in identifying true anomalies remains uncertain. Without labeled ground truth or expert confirmation, we cannot currently be certain of whether the anomalies flagged are genuinely meaningful in an operational context or if they stem from modeling artifacts or sensor noise. Therefore, expert feedback and collaborative evaluation will be essential steps for future work before industrial deployment. This limitation is common in industrial anomaly detection, as also noted in the literature (Amarbayasgalan et al., 2020), where unsupervised methods are often employed due to the scarcity of labeled data. However, without qualitative validation or ground-truth comparisons, it is difficult to assess whether the system is producing true positives or false alarms.

5.1.2 Alignment with Literature and Contribution

The methodology aligns with and draws inspiration from several influential works discussed in the background section. The use of autoencoders for unsupervised reconstruction-based anomaly detection comes from the strategies employed in systems like RE-ADTS (Amarbayasgalan et al. ,2020) and MSCRED (Zhang et al., 2019), which rely on learning the distribution of normal data and using reconstruction error as a signal for anomaly. Similarly, our thermocouple-level regression-based prediction models reflect the predictive strategy seen in works like DeepAnT (Munir et al. (2019a)) and Wu et al. (2022), where deviations between predicted and observed values serve as the anomaly signal.

A key point of comparison with literature, such as studies on MSCRED and MCRAAD, lies in the treatment of temporal relationships. These sophisticated frameworks use architectures like recurrent neural networks to model long-term dependencies. Drawing inspiration from their core principle, that historical context is critical for time series data, this thesis adopted a more direct, interpretable, and computationally efficient approach by incorporating lagged features. By augmenting the input for every model with the state of the system from the previous timestamp (t-1), a crucial layer of temporal memory was introduced. The performance gains observed, particularly the increase in precision for the diagnostic regression models, serve as powerful evidence for the effectiveness of this strategy.

This approach represents a deliberate design choice. Given that this is the first such monitoring system developed for these furnaces, the goal was to establish a robust, interpretable, and efficient baseline. While the current single-step lag is effective at capturing sudden changes and short-term trends, it may not capture more subtle, long-range patterns like gradual sensor drift as effectively as a dedicated recurrent model. Therefore, this work serves as a successful foundational step, showing the importance of temporal data and providing a strong justification for exploring more complex time-aware architectures in future iterations to enhance the system's capabilities further if needed.

5.1.3 Generalizability and Interpretability

In its current form, the proposed architecture is tailored to the specific structure of the furnace dataset, with known zones, thermocouple groupings, and fuel flows. However, the underlying principles are generalizable. Any industrial system with a modular sensor layout and hierarchical physical structure (e.g., turbines, reactors, or smart buildings) could potentially benefit from a similar strategy.

5.2 Ethical and societal aspects

The dataset used in this project was provided by Outokumpu. The data consists of time-series sensor readings from one of their industrial furnaces. All rights to the data are reserved by Outokumpu, and its use in this study was granted solely for academic and research purposes under confidentiality and non-disclosure terms.

The developed anomaly detection framework is intended for process monitoring and decision support, not for automated control or fault prediction without human oversight. Interpretation of results remains within the domain of process engineers and operators. Collaboration with domain experts is recommended for future validation of detected anomalies, calibration of model thresholds, and consideration of real-world deployment constraints.

6 Conclusion

This thesis aimed to design and implement a hierarchical anomaly detection system tailored to the complex sensor environment of an industrial furnace. The primary aim was to enable both broad anomaly detection at the furnace level and detailed diagnosis within specific zones and sensors, using a combination of unsupervised autoencoders and supervised regression models. The anomaly detection results are visualized through dashboards for each timestamp that provides real time charts for each level of anomaly detection; the dashboard also features an overview page where it analyses the trends of anomalies over time by providing useful and actionable charts.

While the system appears robust in its layered design, it was not able to fully answer whether the flagged anomalies correspond to real-world operational issues. This is primarily due to the absence of labeled anomaly data and expert annotations. However, isolation forest, which was used as the benchmark for autoencoder, showed relatively positive correlation on anomaly detection; the true value of these models should however be decided upon real deployment and over time.

By combining reconstruction-based detection and predictive modeling, this work contributes a hybrid strategy that aligns with recent developments in unsupervised anomaly detection literature and is tailored for this specific industrial case, while remaining grounded in practical engineering concerns like interpretability and modularity. When placed in a broader context, it demonstrates the potential of AI-driven monitoring systems in data-rich industrial environments.

6.1 Future work

The project, while yielding promising structure, faced several limitations that offer clear directions for future research and development.

The system’s outputs have not been validated by domain experts. Collaborating with engineers and operators could greatly improve trust in the system, refine its thresholds, and even allow the development of a semi-supervised approach using expert-tagged anomalies. Future iterations should prioritize involving experts in evaluating the system's alerts and refining its behavior through time.

The current system successfully incorporates short-term temporal dependencies by utilizing a single-step lag (t-1), which proved highly effective in improving model performance. However, a natural direction for future work is to enhance this capability to capture more complex, long-range temporal patterns. While the current approach is good at detecting sudden spikes and recent changes, it may be less sensitive to gradual sensor drift or subtle cyclical behaviors that unfold over many hours or days. Future iterations could integrate more sophisticated time-aware architectures. This could involve using a wider window of lagged features (e.g., t-5, t-10, t-60) or implementing dedicated sequential models like Long Short-Term Memory (LSTM) networks or Transformers.

The thermocouple-level diagnostics currently rely on Ridge regression models. While these models proved to be exceptionally precise after the inclusion of lagged features, their linear nature may not capture all possible non-linear interactions between process variables. Future improvements should focus on non-linear regression models like Gradient Boosting Machines (e.g., XGBoost) or Feedforward Neural Networks. These models can learn more intricate relationships, leading to enhanced sensitivity for detecting subtle thermocouple deviations. Future work could test the hypothesis that including features from adjacent zones (e.g., the temperature of a neighboring zone's TC) could provide additional contextual information and create an even more robust and accurate predictive model.

It is also important to note that the level-based, gated implementation described in this thesis represents one possible deployment strategy, designed as a prototype to prioritize computational efficiency for a potential real-time scenario. In this model, zone-level analysis is contingent on a furnace-level anomaly flag. However, this is not the only viable approach. Since the operational analysis does not necessarily require instantaneous real-time processing, an alternative and potentially more robust implementation would be to run all furnace-level and zone-level autoencoders in parallel for every timestamp. This parallel approach would eliminate the risk of a significant but localized zone anomaly being missed by the furnace-level model. While this would increase the computational load, it would provide a more comprehensive diagnostic view at all times. The choice is therefore a practical decision, dependent on the specific computational resources available and the operational tolerance for different types of potential detection misses.

References

Amarbayasgalan, T., Pham, V. H., Theera-Umpon, N., & Ryu, K. H. (2020). Unsupervised anomaly detection approach for time-series in multi-domains using deep reconstruction error. Symmetry, 12(8), 1251.

Munir, M., Siddiqui, S. A., Dengel, A., & Ahmed, S. (2019). DeepAnT: A deep learning approach for unsupervised anomaly detection in time series. IEEE Access, 7, 1991-2005.

Pereira, J., & Silveira, M. (2018). Unsupervised anomaly detection in energy time series data using variational recurrent autoencoders with attention. In 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 1275-1282). IEEE.

Provotar, O. I., Linder, Y. M., & Veres, M. M. (2019). Unsupervised anomaly detection in time series using LSTM-based autoencoders. 2019 IEEE International Conference on Advanced Trends in Information Technologies (ATIT), 513–517. https://doi.org/10.1109/ATIT.2019.8660305

Wu, W., He, L., Lin, W., Su, Y., Cui, Y., Maple, C., & Jarvis, S. (2022). Developing an unsupervised real-time anomaly detection scheme for time series with multi-seasonality. IEEE Transactions on Knowledge and Data Engineering, 34(9), 4147-4160.

Zhang, C., Song, D., Chen, Y., Feng, X., Lumezanu, C., Cheng, W., Ni, J., Zong, B., Chen, H., & Chawla, N. V. (2019). A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In Proceedings of the AAAI Conference on Artificial Intelligence, 33(01), 1409-1416.

Zhang, Y., Chen, Y., Wang, J., & Pan, Z. (2023). Unsupervised deep anomaly detection for multi-sensor time-series signals. IEEE Transactions on Knowledge and Data Engineering, 35(2), 2118-2131. (Note: The year and volume/issue reflect the final publication date, the OCR showed 2021 as publication.)

Zhao, P., Chang, X., & Wang, M. (2021). A novel multivariate time-series anomaly detection approach using an unsupervised deep neural network. IEEE Access, 9, 109025-109039.

IBM. (2020). Anomaly detection. IBM Think. https://www.ibm.com/think/topics/anomaly-detection

IBM. (2021a). What is a neural network? IBM Think. https://www.ibm.com/think/topics/neural-networks

IBM. (2021b). What is unsupervised learning? IBM Think. https://www.ibm.com/think/topics/unsupervised-learning

IBM. (2023). What is ridge regression? IBM Think. https://www.ibm.com/think/topics/ridge-regression

O'Connor, R. (2022, January 3). Introduction to variational autoencoders using Keras: The complete guide to understanding and implementing variational autoencoders with Keras. AssemblyAI Blog. https://www.assemblyai.com/blog/introduction-to-variational-autoencoders-using-keras

TseKiChun. (2021, November 11). Neural network explain.png. Wikimedia Commons. Retrieved from https://commons.wikimedia.org/wiki/File:Neural_network_explain.png

Gupta, B. B. (2022). Autoencoder architecture [Figure 6]. In Identification of Fraudulent Online Transactions and Protection: State-of-art Techniques. ResearchGate. Retrieved from https://www.researchgate.net/Figure/Autoencoder-architecture_fig3_365880792

GeeksforGeeks. (2024, July 15). What is Isolation Forest? GeeksforGeeks. https://www.geeksforgeeks.org/what-is-isolation-forest/

Appendices

The files that are included with this report:

- ModelCreations.ipynb: This Jupyter notebook handles the entire model training process. It prepares the raw data and then trains the autoencoder, Isolation Forest, and Ridge regression models, subsequently saving the trained models for later use.
- Benchmark.ipynb: This notebook is dedicated to model comparison. It benchmarks the performance of the trained autoencoder models against their corresponding Isolation Forest models.
- Analyzer.py: This Python script is designed for operational data processing. It directly ingests CSV files in the company's specified format, processes them row by row, and applies the appropriate autoencoder or regression models. The script extracts all necessary information, saving it in a specific format tailored for consumption by the Streamlit user interface.
- Dashboard.py: This Streamlit application serves as the user interface. It reads from the output CSV file generated by Analyzer.py and presents the results through interactive plots, visuals, and detailed information. The dashboard offers both granular timestamp-level details and a comprehensive summary overview.

Zone	Regressor (TC)	Training samples	Testing samples	Train MSE	Train R^2	Test MSE	Test R^2
ZON1	UgnZon1TempRegAr (TC1)	165 839	41 460	84.6670	0.9975	396.4718	0.9925
ZON1	UgnZon1TempSkyddAr (TC2)	165 839	41 460	85.7411	0.9975	413.4806	0.9925
ZON1	UgnZon1TempVaggOverBandAr (TC3)	165 839	41 460	2 949.3834	0.9036	1 780.5389	0.9588
ZON1	UgnZon1TempVaggUnderBandAr (TC4)	165 839	41 460	1 759.0795	0.9324	1 981.3624	0.9534
ZON2	UgnZon2TempAr (TC1)	165 839	41 460	392.0346	0.9826	742.6428	0.9839
ZON2	UgnZon2TempSkyddAr (TC2)	165 839	41 460	1 568.8243	0.9435	806.3221	0.9843
ZON2	UgnZon2TempVaggOverBandAr (TC3)	165 839	41 460	7 111.5529	0.7711	4 206.6153	0.9181
ZON2	UgnZon2TempVaggUnderBandAr (TC4)	165 839	41 460	555.2175	0.9790	1 095.4044	0.9807
ZON3	UgnZon3TempRegAr (TC1)	165 839	41 460	104.0707	0.9972	627.5676	0.9912
ZON3	UgnZon3TempSkyddAr (TC2)	165 839	41 460	115.9567	0.9969	686.3155	0.9896
ZON3	UgnZon3TempVaggAr (TC3)	165 839	41 460	950.3552	0.9832	1 205.4040	0.9854
ZON3	UgnZon3Temp (TC4)	165 839	41 460	1 794.5122	0.9620	4 970.2558	0.9391
ZON3	UgnZon3Temp (TC5)	165 839	41 460	838.2048	0.9795	981.3866	0.9860
ZON4	UgnZon4TempAr (TC1)	157 040	39 260	1 336.3294	0.9635	307.0338	0.9935
ZON4	UgnZon4TempSkyddAr (TC2)	157 040	39 260	921.7410	0.9736	264.7593	0.9937
ZON4	UgnZon4TempVaggAr (TC3)	157 040	39 260	2 830.4872	0.9357	1 693.3938	0.9724
ZON4	UgnZon4TempVaggAr (TC4)	157 040	39 260	0.0000	1.0000	0.0000	1.0000
ZON5	UgnZon5TempAr (TC1)	165 839	41 460	15.7421	0.9996	15.6751	0.9997
ZON5	UgnZon5TempSkyddAr (TC2)	165 839	41 460	15.7052	0.9996	15.5735	0.9997
ZON5	UgnZon5TempVaggAr (TC3)	165 839	41 460	2 956.9429	0.9286	4 215.9012	0.9544
ZON6	UgnZon6TempAr (TC1)	165 839	41 460	33.1206	0.9992	62.0981	0.9990
ZON6	UgnZon6TempSkyddAr (TC2)	165 839	41 460	32.0307	0.9992	60.5166	0.9990
ZON6	UgnZon6TempVaggAr (TC3)	165 839	41 460	2 242.1270	0.9496	2 433.3705	0.9754
ZON6	UgnZon6TempUtgValvAr (TC5)	165 839	41 460	3 845.5161	0.8848	1 289.4831	0.9808

Table 1. Evaluation of Regression results for thermocouple models before the addition of lag