Results in Engineering 30 (2026) 110483

Contents lists available at ScienceDirect

Results in Engineering

journal homepage: www.sciencedirect.com/journal/results-in-engineering

Research paper

A group-aware temporal framework for quality indicator prediction and anomaly detection in production

Thanos Kontogiannis [a,b,*], Dimitrios Zarouchas [b], Nick Eleftheroglou [a]

[a] Intelligent System Prognostics for Operations and Maintenance Group, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, Delft, 2629 HS, Netherlands

[b] Center of Excellence in AI for structures, Prognostics & Health Management, Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, Delft, 2629 HS, Netherlands

ARTICLE INFO

Keywords:
Group-aware learning
Anomaly detection
Non-i.i.d. data
Temporal dependencies
Coiling temperature prediction
Hot strip mill
Steel manufacturing
Predictive control

ABSTRACT

This work presents a data analysis-based, group-aware framework for predicting quality indicators with anomaly detection in non-i.i.d. datasets that exhibit short temporal dependencies. The design is motivated by statistical diagnostics of temporal autocorrelation and intraclass variance, which highlight the need for causal temporal encoding and group-level decomposition. The framework integrates a residual-boosted regressor, a group-aware anomaly detector, and a calibrated fusion scheme that balances precision and recall. Evaluation is conducted on real production data from hot strip mill operations, with coiling temperature prediction serving as a case study. A key contribution is interpreting coiling temperature dips, previously treated as outliers, as proxies for surface anomalies, thereby enabling their explicit detection. Results demonstrate consistent gains over physics-based and tabular machine learning baselines, confirming that the framework provides more reliable quality-risk indication for decision support in industrial predictive-control workflows.

1. Introduction

Industrial production datasets are rarely independent and identically distributed (i.i.d.). Instead, they are structured in groups defined by production batches, lots, or experimental runs, each influenced by its own operating window, raw-material chemistry, or ambient conditions. Such grouped, non-i.i.d. data are common across diverse domains. In semiconductor fabrication, wafers processed in the same lot share conditions that induce lot-level biases in production output and yield modeling [1]. In high-dimensional omics studies, batch effects stemming from laboratory or sequencing runs systematically shift distributions, requiring explicit correction to preserve biological signal [2,3]. Even in distributed learning settings, the heterogeneity of local clients drives the design of federated optimizers that explicitly account for group-level drift [4,5]. These examples underline a general principle: ignoring group structure risks mistaking systematic biases as random noise, thereby undermining predictive modeling and anomaly detection.

To address such challenges, the machine learning research community has developed a family of group-aware models. DeepSets [6] introduced permutation-invariant pooling to encode group context into predictions, but its high-dimensional pooled representations are difficult to interpret and rely on sufficient within-group sampling, which fails under sparse or imbalanced conditions common in industrial data. Mixed-effects neural networks [7–10] extend deep networks with group-specific parameters analogous to random effects, capturing local offsets but at the cost of scalability and poor generalization to unseen groups. In omics and neuroimaging, ComBat [11] and its deep extensions [12–14] were originally applied as a preprocessing step to estimate and remove group-specific biases, and then used to learn batch-specific affine transformations to standardise intermediate representations across groups. However, they can inadvertently discard a signal of operational importance. Federated learning approaches such as FedProx [5] and FedDANE [15] distribute group-level training across clients, but require persistent, coherent local datasets and incur high storage and communication overheads. While diverse, these methods share three limitations: they either fail to scale effectively to thousands of groups, fail to generalize well to unseen groups, or risk discarding group effects that may themselves carry crucial information.

This paper examines the coiling temperature (CT) prediction problem in the steelmaking hot strip mill (HSM) process as a paradigmatic case of grouped, non-i.i.d. industrial data. The CT, defined as the strip temperature immediately before coiling, reflects both the forming history in the finishing mill and the thermal history in the run-out table (ROT). Within each strip, spatially adjacent locations along the length

* Corresponding author.
E-mail address: a.kontogiannis@tudelft.nl (T. Kontogiannis).

https://doi.org/10.1016/j.rineng.2026.110483
Received 3 February 2026; Received in revised form 20 March 2026; Accepted 8 April 2026
Available online 14 April 2026

Nomenclature

τ_{QI}	Quality indicator threshold
τ_{anom}	Anomaly probability threshold
ACF	Autocorrelation function
AP	Average precision
BCE	Binary cross-entropy
CNN	Convolutional neural network
CT	Coiling temperature
CV	Cross-validation
FCN	Fully-connected network
F_1	F_1-score (harmonic mean of precision and recall)
F_2	F_2-score (recall-weighted F-measure)
Fβ	F$_\beta$-score (generalized F-measure)
GA	Genetic algorithm
GBDT	Gradient-boosted decision tree
GPU	Graphics processing unit
HSLA	High-strength low-alloy
HSM	Hot strip mill
ICC	Intraclass correlation coefficient
i.i.d.	Independent and identically distributed
IR	Infrared
MAE	Mean absolute error
MIG	Multi-Instance GPU
MLP	Multilayer perceptron
MPC	Model predictive controller
MSE	Mean squared error
NN	Neural network
OOF	Out-of-fold
PR	Precision-recall (curve)
R^2	Coefficient of determination
RMSE	Root mean square error
ROT	Run-out table
VRAM	Video random-access memory
XGBoost	Extreme Gradient Boosting

are sampled sequentially as the strip moves past the pyrometer, so spatial correlation along the strip appears as temporal autocorrelation in the recorded CT sequence. Across strips, minor cumulative differences in microstructure, tramp element content, and equipment condition induce systematic shifts in CT level and variability, resulting in each coil forming a distinct group. It is well-established [16–20] that the CT holds information about the achieved microstructure of the steel and thus, is indicative of the material's end properties. Accurate CT modeling is therefore crucial to process control, as deviations from the goal CT profile result in scrap, reduced yield, and degraded material properties.

Over the years, CT prediction has been pursued through increasingly complex analytical and numerical approaches. Early static heat-balance formulations [21] approximated the temperature drop as a function of water flow density, line speed, and strip thickness, yielding actionable predictions for feed-forward control that significantly improved production yield and material quality. While this demonstrated applicability, the aforementioned model can be considered crude and is likely to underrepresent the process. This is to be expected, since the cooling of the material is a complex thermal transfer process involving air and water convection, the Leidenfrost effect, radiation, and phase-transformation-induced heat generation, all of which are dynamic and temperature-dependent. Thereby, the need for more advanced modelling schemes emerged that can better capture the intricacies of the aforementioned process. These included recasting heat-conduction equations into Hamiltonian systems [22], implementing finite-difference solvers with Kalman filtering [23], or fitting temperature-indexed splines to large coil datasets [24]. While significant and improving, all rely on simplifying assumptions, fixed heat-transfer coefficients, or recursive updating, which can lead to small biases that accumulate and cause sizable drifts.

The increasing availability and volume of production data have motivated the development of multiple data-driven alternatives. Xie et al. [25] combines a linear-regression heat-balance core with a small back-propagation neural network (NN) that learns to correct the residuals, Li et al. [26] introduces a least-squares support-vector machine whose hyperparameters are tuned by a genetic algorithm (GA), Sun et al. [27] also utilizes GA for tuning the weights of a feed-forward NN. In contrast, Liu et al. [28] trains a simple three-layer NN on plant laminar-cooling data. Hu et al. [29] builds a data-driven finish-entry-temperature model using a shallow NN optimised by the Grey-Wolf meta-heuristic, Chen et al. [30] deploys a bi-directional long-short-term memory recurrent NN to predict the full temperature field of variable-velocity coils, and Panjiar and Murugananth [31] trained a feed-forward NN specifically on dual-phase steel coils. More recently, Zhang et al. [32] proposed a hierarchical feature-fusion-based CT predictor that combines GRU-attention modeling of continuous strip-level variables, SlowFast-style extraction of discrete header-status signals, and SENet-based coil-level feature fusion for strip-level and next-coil prediction. These approaches consistently reduce prediction error by 15–30% relative to physics-based baselines reporting errors in the range from a $\pm 10\,^{\circ}\mathrm{C}$ band to sub-6 mean absolute error (MAE) [30]. Yet they also share a critical blind spot: abrupt dips in CT profiles below the lower acceptable temperature bound, most often seen in heavy-gauge coils, are routinely discarded during preprocessing. Labeled as outliers, these dips are clipped or smoothed, and the models are trained only on smooth cooling trajectories.

By studying these temperature dips alongside corresponding optical images that reveal surface cracking, we present evidence that these dips are not stochastic noise, but rather strong indicators of surface defects that compromise product quality. Treating them as noise excludes precisely the events most relevant for predictive diagnostics and downstream process control. Consequently, we reformulate CT modeling as a dual task: (i) regression of the continuous CT profile for integration into ROT control and downstream tuning, and (ii) anomaly detection of dips crossing the lower specification bound, directly linked to surface quality failures.

To address the dual-task challenge in a non-i.i.d. industrial setting, we propose a data-analysis-guided, group-aware framework. Exploratory variance ratio diagnostics reveal that between-coil variation dominates within-coil variation, motivating group-aware architectures. Autocorrelation analysis highlights short-range dependencies, justifying the use of causal convolutional filters. Guided by these insights, we decouple the tasks: a residual-boosted neural regressor combines gradient-boosted trees with a compact multi-layer perceptron (MLP) trained on residuals, yielding robust predictions against analytical and boosted baselines. The learned regression trunk is then used to initialize a group-aware causal convolutional classifier, trained with balanced group-aware sampling and focal binary cross-entropy (BCE) loss to detect rare dips. Finally, a calibrated fusion scheme combines regression margins with anomaly probabilities, optimizing the F_1 score under fixed operating thresholds. The comparison space for this framework is therefore centered on industrial, tabular, and structured deep-learning baselines that match the supervised, grouped, and temporally local nature of the problem, rather than on generic unsupervised anomaly-detection methods developed for unlabeled i.i.d. outlier discovery.

Evaluated on production data from Tata Steel Nederland B.V. (TSN), the framework improves CT regression relative to both physics-based and boosted-tree models, while enabling the detection of quality-critical CT dips. Ablation studies confirm the contributions of group-aware offsets, causal convolutions, and weight transfer. Beyond the case study, the framework exemplifies how industrially motivated data analysis can safeguard against data leakage and guide the design of interpretable, group-aware architectures for non-i.i.d. settings.

The contributions of this work are as follows:

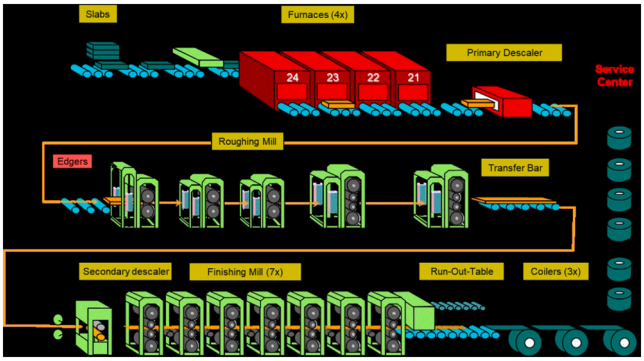

Fig. 1. Overview of the HSM process.

- Evidence that abrupt dips in CT profiles are indicative of surface defects and should be explicitly modeled rather than discarded.
- A data analysis-based, group-aware framework grounded in autocorrelation and variance-ratio diagnostics.
- A residual-boosted neural regressor for CT profiles, robust and leakage-safe.
- A group-aware causal convolutional anomaly detector, combining short-range temporal encoding, coil-aware decomposition, and transferred trunk initialization for rare anomaly detection under a balanced sampler and focal BCE loss.
- A calibrated fusion scheme that integrates regression margin and anomaly probability to improve rare-event detection under the F_1 score criterion.

The rest of this paper is organized as follows. Section 2 presents the process context, the link between CT profiles and surface defects, and the exploratory data analytics that guide design. Section 3 details the models and training pipeline, including regression, classification, and fusion. Section 4 reports experimental results, coil-level case studies, and ablations. Section 5 discusses broader implications and limitations, and Section 6 concludes.

2. Process context - Dataset description

The HSM process, as presented in Fig. 1, transforms reheated steel slabs that are hundreds of millimeters thick into long strips of a final thickness of a few millimeters through a sequence of rolling stands. After descaling and roughing, the strip enters the finishing mill, a series of closely spaced stands that progressively reduce thickness and elongate the strip at high speed. The thermal and mechanical history accumulated during finishing directly influences the evolution of microstructure and the downstream properties of the product.

Once exiting the last finishing stand, the strip passes onto the ROT, where it is transported at controlled speed toward the coiler. The ROT is a long stretch of a series of water spray headers arranged across multiple cooling zones, both at the top and bottom faces of the steel strip. Each header can be activated with variable flow rates and spray patterns, providing local thermal control. In addition, the transport speed across the ROT determines the residence time before coiling and thus the overall cooling trajectory. The combined effects of the finishing mill process, exit temperature, cooling water distribution, and transport dynamics define the CT profile along the length of the strip.

The CT is a critical process variable: it links upstream hot rolling conditions with downstream coil properties such as mechanical strength and phase composition. To achieve the target properties of the material, the ROT process is actively controlled based on the expected CT value predicted before the material enters the cooling zone. Thus, accurate CT prediction is of paramount importance for achieving the desired material properties.

2.1. Surface quality & CT profiles

Surface defects can appear during manufacturing, significantly diminishing the end surface quality of the manufactured steel strips. Known root causes of surface-quality defects include material defects, process defects, and corrosion defects [33]. Material and process defects can be more easily avoided by tailoring composition and controlling manufacturing disturbances (i.e., timely inspection and replacement of rollers). Corrosion defects are, by nature, more challenging. The low stability of the typical three-layer oxide composition of steel (hematite Fe_2O_3, magnetite Fe_3O_4 and wustite $Fe_{1-y}O$) at the low CTs, the presence of other elements in low-carbon steel, the presence of inclusions, the continuous cooling conditions, the temperature gradient across the width of the strip, the absence or lack of oxygen in the centre regions of the coil, the deformation in the mills and the tension between each reduction phase, affect the oxide evolution [34,35].

The extensive study of Min et al. [36] revealed a correlation between the thickness of the oxide layer and the surface quality. This is attributed to the fact that a thicker oxide scale is more brittle and, thus, more prone to chipping off. As demonstrated in the same study, measuring the oxide layer thickness during production is not feasible. Production must be halted, and the formation of the oxide layer must be frozen (i.e., by spraying molten glass on the surface). This process can quickly become costly and counterproductive for a real-world application. This results in uncontrolled oxide behaviour, which, as mentioned earlier, causes critical surface defects. These defects are not only unforeseen but, because their root causes cannot be directly controlled, there is currently no chance of mitigating them. This highlights the need to predict them indirectly.

When plotting the CT across the length of the produced coil, it should be smooth and close to the target temperature as observed in Fig. 2a. During production, two to three bottom-side optical images are captured at the coiling station for quality control. The images are a snapshot of 8m of the strip length and are taken approximately 50 m after the head (beginning) of the strip and 9 m before the tail (end), as depicted in Fig. 2b. It can be seen that, in addition to the CT drops at the head or tail of the coil, which are expected in the HSM process, the measured CT closely tracks the goal CT profile, and no surface defects are observed.

In Fig. 3, however, two examples of the same heavy-gauge steel can be seen, whose CT presents significant dips that overshoot the lower CT bound as defined by the recipe of the steel grade. Looking at the respective optical images, they show major surface defects near the centerline of the strip. The CT is measured with infrared (IR) pyrometers, also on the centerline of the strip. Granted that IR measurements are strongly affected by material emissivity, it is evident that the dips observed in the CT plots are not merely measurement noise or corrupted data. Instead, this study proposes that the dips are caused by severe surface defects resulting from the corruption of the oxide layer. In this case, the oxide layer no longer covers the entire strip surface, leading to an apparent underestimation of temperature due to emissivity variation.

Therefore, it is proposed that significant dips in the CT profiles can serve as proxies for these major surface defects and should not be discarded during preprocessing or modelling of the process. Instead, they should be directly predicted utilizing the data available preceding the ROT process.

While optical images or videos taken at the coiling station (as shown in Fig. 3) could be used to directly detect surface defects using computer vision techniques, this would require hundreds of high-resolution photos for each produced coil. This demands significant resources, not only for handling and storing the data but also for developing a model for optical detection. Additionally, as shown in Fig. 4, it is pretty common for the strip to end up with an irregular shape, creating high and low spots that leave part of the strip out of focus. Finally, while promising for detecting surface defects at the coiling station, any optical approach directly on the coiling station lacks predictive capabilities, serving only as a quality-control feature to reject affected material. That is why the

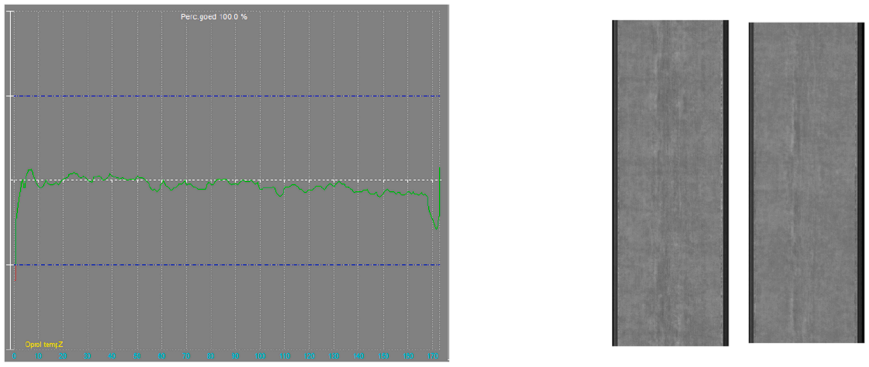

Fig. 2. Example of a good quality coil: (a) measured CT plot across the length of the strip (blue dashed lines annotated the upper and lower acceptable CT bounds), and (b) optical images 45 m from the head (left) and 9 m before the tail (right). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

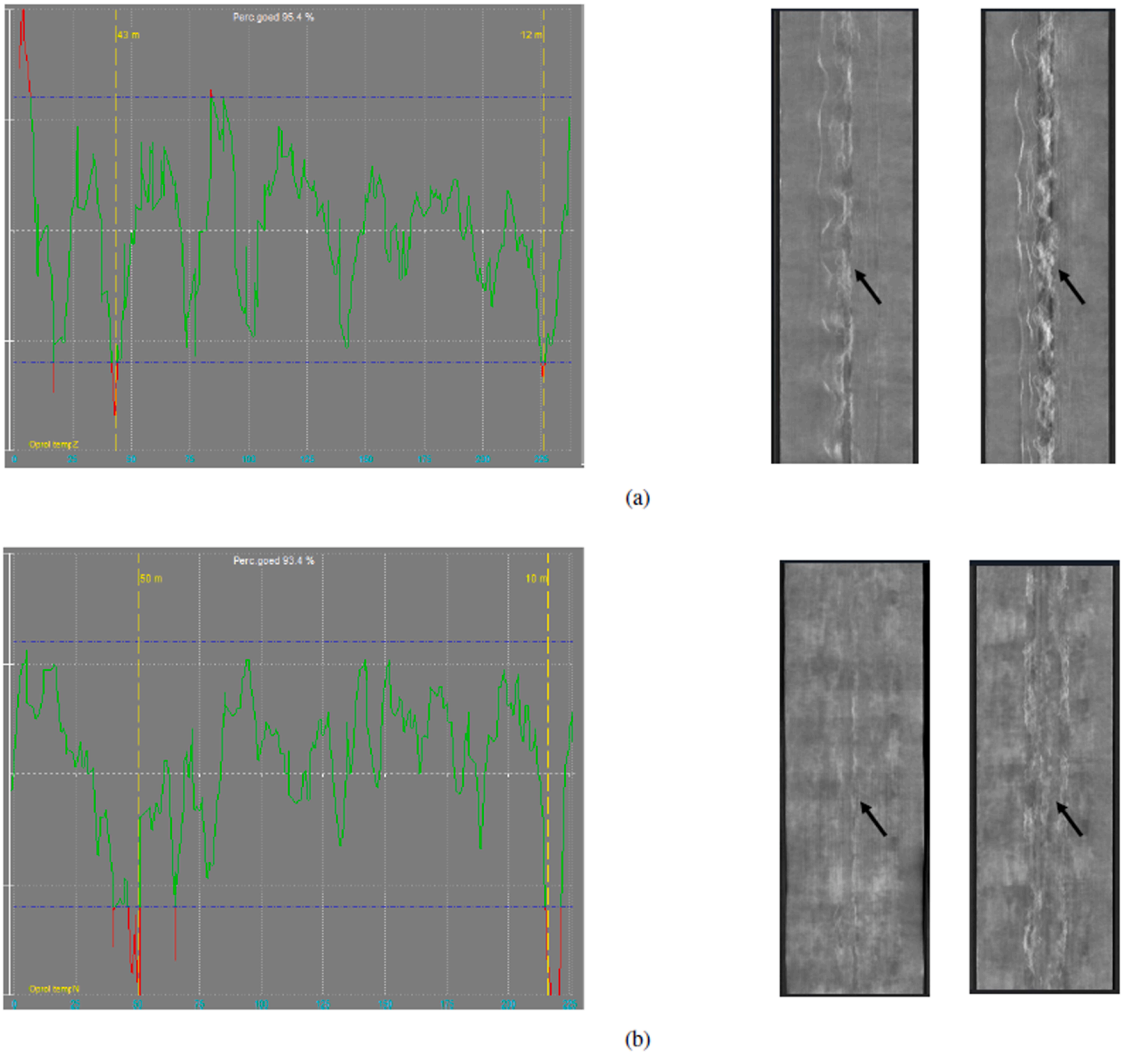

Fig. 3. Example strips with major surface defects, measured CT profile on the top and bottom left, and corresponding optical images taken at the annotated yellow dashed lines. Black arrows point to the major surface defects on the centerline of the strip. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

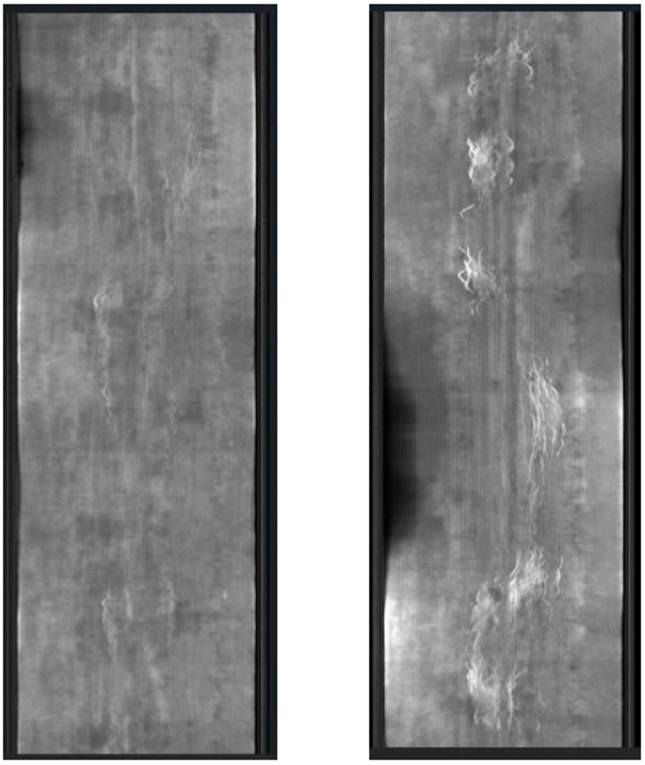

Fig. 4. Steel strip with irregular shape causing part of the image to be out of focus.

current study proposes utilizing the available data directly before the ROT process. That way, the aforementioned dual-task framework achieves the following: an accurate and early CT prediction, as well as the early prediction of surface defects, thereby enabling proactive control of the finishing mill and ROT parameters, which mitigates defects.

2.2. Data and problem definition

The dataset available for this study comprises production data from Tata Steel Nederland B.V. (TSN) covering 2018–2022 for a high-strength low-alloy (HSLA) steel grade commonly produced for heavy-gauge strips (as shown in Fig. 5a, where most of the produced coils are thicker than 8mm). After filtering out the coils with missing or corrupt data, the remaining considered coils number approximately 8000. The processing parameters, such as rolling forces, velocity, interstand cooler activity, descaler activity, and water flow setpoints on the ROT, are included in the dataset. To facilitate an early prediction of the CT, only the available process parameters directly preceding the ROT process are used for modeling. This is why only setpoints and not measurements of the water flow on the ROT are included in the dataset.

2.2.1. Indexing, synchronization, and scaling of CT values

The CT is measured with an infrared pyrometer at the centerline of the bottom side of the steel strip. The CT is recorded and saved at 1 m intervals along the strip's length. The process parameters, on the other hand, are more challenging to index consistently, as the material elongates and moves at different velocities throughout the various stages of the HSM process. This would mean that if the process parameters were recorded at set time intervals, they would not necessarily refer to the same material. To tackle this challenge, the initial steel slab is split into 50–70 segments that track the same material throughout the process, accounting for elongation and velocity changes. Each segment corresponds to a specific length on the final strip at the coiling station. Therefore, the first step in preparing the dataset was averaging the multiple CT measurements per segment to obtain a single CT value.

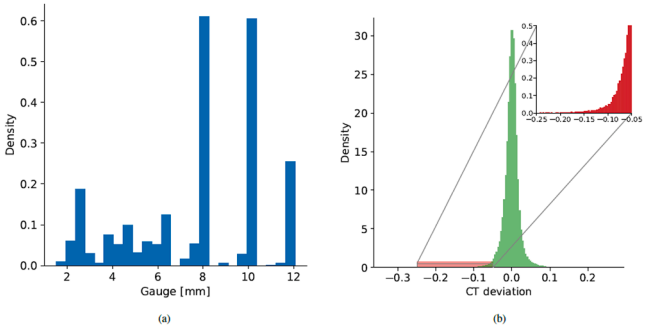

Fig. 5. Density histogram plots for: (a) Gauge of the steel grade, and (b) Scaled CT values. The red vertical line annotates the -0.049 value below which the anomalies lie. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Next, since the CT values are in the several hundred degrees Celsius range, they need to be scaled to lower values to avoid vanishing gradients during NN model training [37,38]. To avoid losing the physical meaning of the CT values or distorting the shape of the profiles [39], a domain-specific scaling scheme is devised and presented in the following:

$$CT_{norm} = \frac{CT_{meas} - CT_{goal}}{CT_{goal}}, \tag{1}$$

where CT_{goal} is the target CT as defined by the recipe of the steel grade and CT_{meas} is the measured CT by the pyrometer. That way, the initial shape of the CT profiles is maintained, and the values are scaled to the goal CT values, which are known a priori, thereby avoiding data leakage.

Regarding the features, standardization (or z-score scaling) is applied to each feature separately to account for different value ranges and aid NN model training.

2.2.2. Targets and labelling

As described in Section 2.1, it is vital to not only predict the values of the CT but also to capture the temperature dips below the allowed lower temperature bound as defined by the target recipe. For that reason, utilizing Eq. (1), the lower specification bound is defined by the recipe of the steel grade under consideration as $\tau_{QI} = -0.049$, which equals a negative 4.9% deviation from the goal temperature. Values below τ_{QI} are labelled as anomalous so that the ability of the models to capture them can be evaluated. The distribution of the CT values is displayed in Fig. 5b. It can be seen that the anomalous values (below -0.049) constitute the left tail of the distribution, and they account for less than 1.5% of the total CT values, classifying them as rare anomalies and making their modeling challenging.

2.2.3. Splits and leakage control

Due to the rarity of the anomaly data, as previously shown, if a random train-validation-test split is performed, no anomaly data will likely end up in the test or validation sets. This can harm accuracy and also lead to a misleading performance evaluation of the models [40]. For that reason, we employ a stratified group split procedure that preserves anomaly prevalence while preventing leakage across coils.

Utilizing the binary anomaly label obtained in Section 2.2.2, an entire coil is deemed anomalous if it contains at least one anomalous value that does not belong in the first or last 5 measurements. That is because temperature abnormalities at the head (beginning) or tail (end) of the coil are common during the HSM process and do not indicate a surface defect. That way, each coil obtains a normal or abnormal label. Using this label, we perform a 5-fold StratifiedGroupKFold on the full dataset, stratifying on the constructed coil label and grouping by coil identifier, ensuring that parts of the same coil do not appear in more than one split. Within each outer training fold, validation is obtained by a second StratifiedGroupKFold on the training portion only, again stratifying by the

Table 1
Normal and abnormal data counts and anomaly ratios for the stratified train-validation-test split.

Subset	Normal Data	Anomalous Data	Anomaly Ratio[%]
Training	300,744	4542	1.51
Validation	69,060	1014	1.47
Test	89,874	1262	1.41

coil label and grouping by coil identifier. This produces three disjoint sets of coils per outer fold - inner-train, validation, and test - with the anomalous-to-normal ratio closely matched across splits, as presented in Table 1.

We adopt K = 5 folds for both train/test and train/validation splits, yielding an effective split of approximately 64/16/20 for inner-train, validation, and test. For realistic results evaluation, the mean and standard deviation across the K cross-validation (CV) folds are reported for each model.

Crucially, coil identity is preserved end-to-end. The coil IDs are passed as grouping variables when forming the splits and carried alongside the features in the data loaders. For coil-aware architectures, each batch includes the coil ID, allowing the model to condition on coil-specific effects, whereas baseline models ignore this field.

2.3. Data analytics - Guiding architectural design

To guide the design of the framework, an exploratory data analysis is recommended for the dataset at hand. The objective of this analysis is to identify the temporal structure and group-level variance in the signals, thereby determining which mechanisms a predictive model must capture to model the process accurately. By examining autocorrelation patterns and variance ratios, direct evidence is obtained on whether temporal dependencies exist, for how long they persist, and whether significant grouping occurs, thereby justifying a group-aware modelling approach. These diagnostics ensure that the model design is not ad hoc but grounded in the statistical properties of the production data. For demonstration purposes, these diagnostics are performed on the aforementioned historical production dataset for the CT prediction task.

2.3.1. Temporal dependencies

A first step in the exploratory analysis is to quantify temporal dependencies in the dataset. This makes sense in production datasets where the data is collected sequentially for each group. The autocorrelation function (ACF) measures the strength of the correlation between each data point and its preceding ones. This reveals the extent of the temporal context a model should incorporate: persistent correlations with longer lag times suggest the need for models with long-term memory, while rapid decay implies short-range time dependency.

The first step is to perform an autocorrelation analysis of the target variable. This can directly indicate whether the modelled phenomenon retains memory. Observing Fig. 6, it can be seen that the average ACF value for the target variable, in this case the CT, across coils shows high correlation at lag 1 that remains statistically significant through lag 2 and then falls within the confidence band. The dashed bounds in the figure correspond to an approximate 95% interval of $\pm 1.96/\sqrt{N}$ under the null hypothesis that the true autocorrelation at all nonzero lags is zero, where N is the effective series length. Here, N was taken as the median number of rows per coil ($\simeq 60$), which provides a conservative bound when averaging ACFs across coils of differing lengths. Intuitively, values inside the band are consistent with no meaningful correlation, whereas values outside indicate significant dependence beyond what would be expected by chance.

The implication for modeling is that the target value exhibits short-range memory of only a few consecutive data points. Anomaly detection should therefore exploit short causal context rather than treating rows independently, and compact convolutions through time spanning

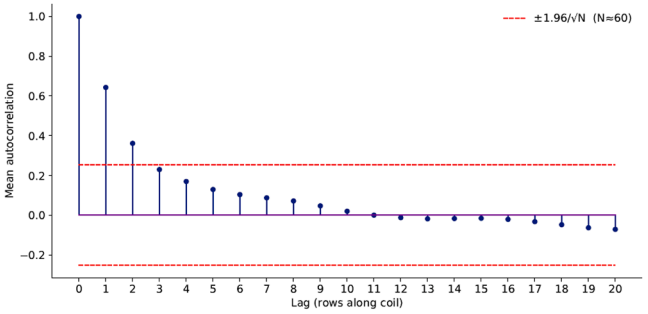

Fig. 6. Average autocorrelation function of coiling temperature across coils with 95% confidence bounds ($\pm 1.96/\sqrt{N}, N \simeq 60$) annotated with the red dashed lines. Significant correlation persists up to lag 2, indicating short-range temporal dependence. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

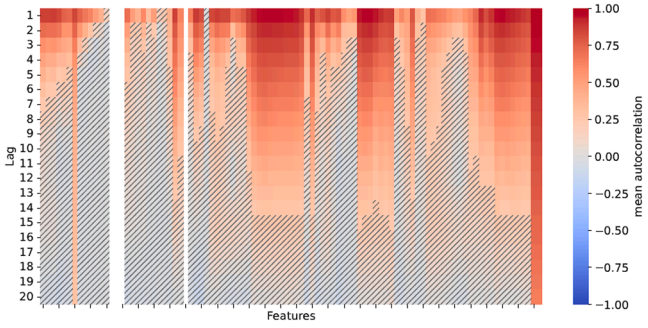

Fig. 7. Heatmap of per-feature autocorrelation functions (ACFs). Each vertical bar represents a feature, and the vertical axis indicates lag in rows. Colour intensity indicates the strength of the correlation, and hatched regions mark values within the 95% confidence bounds ($\pm 1.96/\sqrt{N},\ N \simeq 60$). Features exhibit heterogeneous temporal persistence, from long memory to near temporal independence. The names of the features have been redacted, adhering to data confidentiality. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

roughly 3–5 rows should be sufficient to capture meaningful dependence without resorting to deep recurrent memory.

The next step is to determine whether this temporal structure is confined to the target or also present in the modelled process's drivers. To achieve that, we computed per-feature ACFs and summarized them as a feature-by-lag heatmap presented in Fig. 7. This feature-by-lag heatmap visualises the ACF values of all engineered inputs. Each vertical bar corresponds to a single feature, while the vertical axis represents lag in rows along the coil. Colour intensity shows the magnitude of the average autocorrelation at that lag, so persistent warm colours indicate long memory and rapid fading indicates short memory or none at all. Hatched regions mark values that fall within the 95% confidence bounds, computed with the same procedure explained for Fig. 6. The figure reveals a heterogeneous landscape: some features retain significant autocorrelation over long time lags, others decay after only one or two rows, and others remain within the confidence band throughout. This pattern implies that the model should account for varying temporal persistence across channels. Using a universal temporal window across all features would inject noise from channels with no memory, but hard-coding exclusions based on the heatmap would risk discarding useful signals. Instead, per-feature gating mechanisms are expected to allow the network to learn which inputs benefit from temporal context and which should bypass it, preserving useful paths while suppressing noise.

2.3.2. Group-aware architecture diagnostic

The temporal analyses above established short-range dependence within the series. The next question is whether systematic level shifts

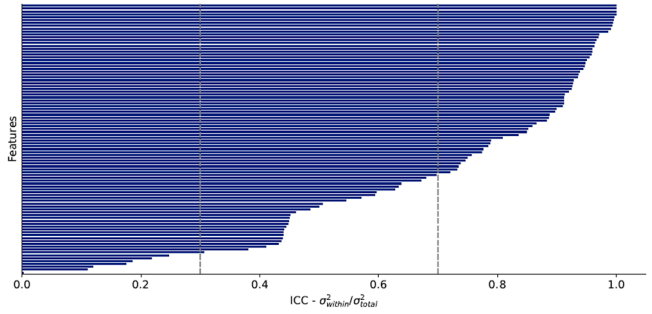

Fig. 8. Intraclass correlation coefficients (ICCs) of all numeric features, sorted from low to high. High values reveal strong group-level (coil-level) grouping, motivating group-aware modeling and underscoring the need for group-level train-test splits to avoid data leakage.

exist across groups, which would further violate the i.i.d. assumption and motivate a group-aware treatment. We quantify this utilizing the intraclass correlation coefficient (ICC) computed per feature under a one-way random-effects model with groups defined by coils and unequal group sizes. Let $\sigma^2_{\text{between}}$ and σ^2_{within} denote the between- and within-group variances calculated from standard ANOVA [41,42], the ICC is defined as:

$$\text{ICC} = \frac{\hat{\sigma}^2_{\text{between}}}{\hat{\sigma}^2_{\text{between}} + \hat{\sigma}^2_{\text{within}}}, \qquad \text{ICC} \in [0, 1]. \tag{2}$$

The full ANOVA details and variance-components derivations are provided in Appenidx A. Intuitively, the ICC measures the proportion of total variance attributable to systematic differences between groups. Values close to zero indicate that variability is almost entirely within groups, while values near one indicate that groups explain most of the variability. Intermediate values reflect partial grouping effects. Therefore, features with high ICC values demand group-aware modeling. This decomposition provides a principled justification for embedding group-awareness into the framework. In the present work, groups correspond to coils, and the target variable is the CT.

Fig. 8 reports the ICC for every feature available for CT prediction, sorted from low to high, with dashed reference lines at moderate (left) and strong (right) effect levels. A substantial fraction of features concentrates well above the midrange, with many approaching unity, confirming that, in this example, group-level (coil-level) offsets dominate their variability. At the same time, a small tail of low-ICC features is present, indicating channels whose behavior is largely invariant across coils.

This pattern supports the need for a group-aware framework. High-ICC channels motivate coil-aware mechanisms. Low-ICC channels should not be forced through the same conditioning and can be allowed to bypass it, thereby preserving information without injecting coil-specific noise. The strong grouping also highlights the importance of how data are partitioned. If train and test sets were split at the row level, parts of the same coil could appear in both sets, allowing group-specific biases to leak into training. This would artificially inflate test performance and mask the actual generalization error. For this reason, all splits for datasets with strong grouping should be performed strictly at the group level. This rationale was followed in this work as presented in Section 2.2.3

The exploratory analyses in this section established ways to identify two central characteristics of the dataset at hand: temporal dependence and grouping. Applying this to the use-case example of the HSM dataset for predicting CT and corresponding anomalies, short-range temporal dependence in the target variable and heterogeneous autocorrelation across features were found, along with strong group-level effects revealed by high ICC values. These findings suggest that predictive models must capture local temporal context without relying on long-term memory, while also conditioning on group identity to prevent leakage and accurately handle between-group variance.

3. Methods and models

Building on the discovery of temporal correlations and grouping at the production data at hand, a framework is proposed to address a dual task: (i) regression of a continuous quality indicator, which is essential for guiding downstream process models and control, and (ii) anomaly detection, focused on identifying rare deviations that are typically treated as noise but may carry critical quality risk, as is the case in the presented use case. As illustrated in Fig. 9, the two tasks are solved by distinct but connected models, a hybrid GBDT-NN regressor and a group-aware short-term temporal classifier. Its design is tailored to exploit short temporal dependencies and strong group-level effects revealed by the data analysis, making it well-suited for the rare-event anomaly detection problem. In the present work, the framework is demonstrated for predicting CT profiles and associated dips in an HSM dataset; however, the methodology applies to a broad class of grouped, non-i.i.d. industrial problems.

3.1. Regressor - Residual-boosted MLP

The regression component of the framework, presented in Fig. 10, is built as a dual-module design: a GBDT baseline combined with an MLP trained on its residuals. This architecture leverages the complementary strengths of tree ensembles and NNs. Tree models provide robust approximations with strong performance on structured tabular data [43,44], while NNs offer flexible function approximation and the ability to transfer learned representations across tasks [45–47]. By stacking them in a residual fashion, the GBDT supplies a stable baseline prediction while the MLP focuses on modeling the remaining structure in the errors. This residual boosted regressor yields both accurate continuous predictions and a transferable trunk that later initializes the anomaly detector.

3.1.1. Out-of-fold predictions

A critical step in residual modeling is generating unbiased baseline predictions. If the GBDT model was simply trained on the entire training set and then used to compute residuals for training a downstream predictor, those residuals would reflect errors conditional on having already seen those observations during training. Such in-sample residuals are optimistically biased: they appear smaller and less structured than the errors the model makes on unseen data. It is important to emphasize that this leakage problem persists even when the GBDT model is never exposed to the held-out test set. The bias is not due to test contamination but to reusing training-fitted predictions as residuals. Training the NN on these biased residuals would lead it to correct noise and spurious patterns specific to the training set, reducing its generalizability, leading to overly optimistic evaluation, and poor out-of-sample performance.

To avoid this, the boosted model is cross-fitted to produce OOF predictions. In K-fold cross-fitting, each fold is held out in turn, and its predictions are generated by a model trained only on the remaining $K - 1$ folds. The concatenated OOF predictions across all folds form a baseline in which each training sample is predicted by a model that did not see it during fitting. The corresponding OOF residuals are therefore unbiased estimates of actual generalization error.

This procedure ensures that the neural residual model is trained only on genuine, out-of-sample mistakes of the boosted ensemble. It prevents the NN from learning to "correct" artifacts of the training set, avoids optimistic bias in evaluation, and provides a solid foundation for capturing systematic error structure beyond the boosted trees.

3.1.2. MLP residual model

Once unbiased OOF residuals are obtained, they are modeled with a multilayer perceptron (MLP) [48]. The MLP complements the GBDT by

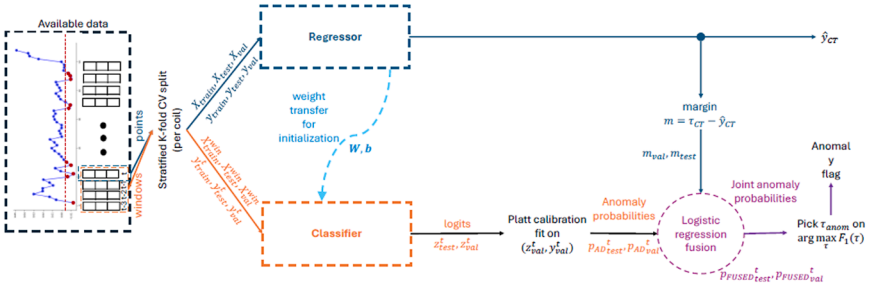

Fig. 9. Proposed framework. The regressor and classifier are presented in detail in Figs. 10 and 11, respectively.

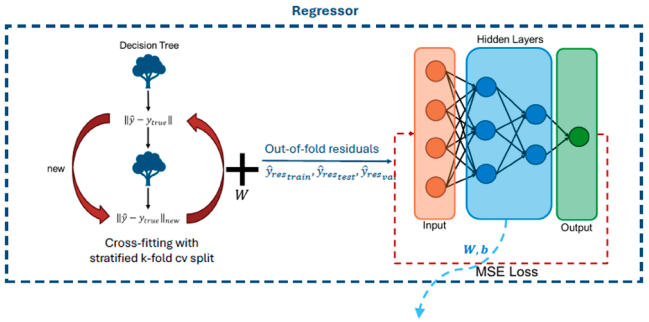

Fig. 10. The regressor of the proposed framework in detail. The blue dashed arrow denotes the weight transfer for the initialization of the classifier described in Section 3.2.3. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

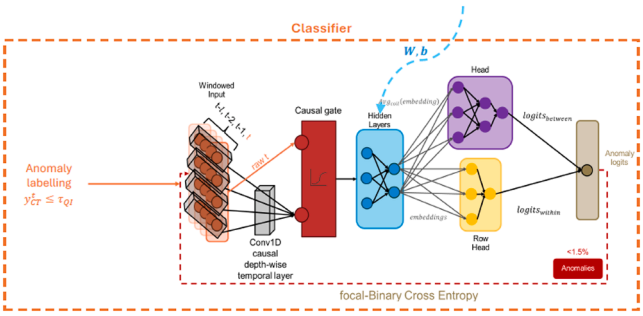

Fig. 11. The classifier of the proposed framework in detail. The blue dashed arrow denotes the weight transfer for the initialization of the classifier described in Section 3.2.3. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

capturing smooth nonlinear corrections and high-order interactions that are difficult for tree ensembles to represent [49,50]. It is structured with an input layer matching the feature dimension, followed by a hidden trunk of fully connected layers that learn a compact residual representation. The final layer consists of a single linear output node, optimized with mean squared error (MSE) loss, to predict the residual correction. This hidden trunk forms the transferable component: after training on residuals, its weights are reused to initialize the anomaly detector, ensuring that the classifier starts from a representation already tuned to the systematic structure of the target variable, as discussed in Section 3.2.3.

The final regression estimate combines the two stages:

$$\hat{y}(x) = \hat{y}_{\mathrm{GBDT}}(x) + \hat{r}_{\mathrm{MLP}}(x), \tag{3}$$

where $\hat{y}_{\mathrm{GBDT}}(x)$ is the OOF baseline prediction and $\hat{r}_{\mathrm{MLP}}(x)$ is the residual correction.

This residual-boosted design improves stability and interpretability: the tree stage anchors predictions to a robust, tabular baseline, while the neural stage learns flexible refinements. Importantly, the MLP trunk trained in this setting yields a transferable representation that is later reused to initialize the anomaly detector, providing continuity between the regression and classification tasks. The following subsection, therefore, turns to the classifier architecture, where this initialization is exploited for anomaly detection.

3.2. Classifier - Coil-aware causal 1D CNN

The second component of the framework presented in Fig. 11 is a classifier designed to detect rare anomalies while accounting for the strong group effects identified in Section 2.3.2. The architecture combines short-range temporal modeling with an explicit decomposition of group-level and within-group variability, ensuring that both systematic group offsets and row-level deviations contribute to the decision boundary. As illustrated in the left side of Fig. 9, the model accepts windowed feature inputs. Then, referring to Fig. 11, the classifier extracts temporal context through causal convolutions and adaptively fuses them with raw features by means of a learnable gating mechanism. The resulting embeddings are passed through the MLP trunk, which is initialized from the regression task, thereby promoting knowledge transfer across objectives. To reflect the non-i.i.d. structure of the data, the embeddings are then decoupled into group-level averages and residual components, which are used to produce centered and per-row logits. These are summed to yield the final anomaly probability. The following subsections provide a detailed description of each stage of the classifier.

3.2.1. Temporal encoder

The first stage of the classifier introduces local temporal context by processing windowed inputs through a depthwise 1-D convolution. The kernel length is set equal to the chosen lag plus 1 to include the current row, corresponding to the effective memory horizon identified in Section 2.3.1. To preserve causality, each prediction is based on a fixed-length window of past rows only, so that the convolution never has access to future values. We therefore apply a depthwise Conv1d with kernel size equal to the chosen lag and no padding on these past-only windows. No dilation is used, since the autocorrelation analysis showed that dependencies vanish within a short range, making contiguous kernels sufficient.

Formally, for an input feature channel $x_t \in \mathbb{R}$ at row t the causal convolution produces an encoded representation

$$h_t = \sum_{k=0}^{L-1} w_k \, x_{t-k}, \tag{4}$$

where L is the kernel length (lag), w_k are the convolution weights. Because the convolution is depthwise, each feature channel is filtered independently, preserving channel specificity while introducing temporal context.

Short causal convolutions are advantageous both statistically and computationally. Statistically, they prevent noise accumulation from irrelevant long-range dependencies. Computationally, they are less time- and data-intensive than recurrent models and can be efficiently

parallelized across channels and samples, avoiding the sequential bottlenecks of long recurrences.

This temporal encoder, therefore, transforms raw feature sequences into embeddings that summarize each row's immediate history. Thereby, the encoder ensures sensitivity to transient deviations while avoiding unnecessary complexity.

3.2.2. Learnable gated fusion

A learnable gated fusion is introduced to regulate the contribution of short-context temporal convolutions on a per-channel basis. The mechanism operates by a convex combination between the current raw feature vector and the corresponding output of a depth-wise causal convolution computed over a short receptive field. Before mixing, both branches are independently normalized to ensure comparable scales. A channel-wise gating vector in $(0,1)^C$ then assigns the relative weight of each branch. The transformed signals, gate, and fused signal are defined as

$$\tilde{x}_t = \mathrm{LN}_{\mathrm{raw}}(x_t), \qquad \tilde{c}_t = \mathrm{LN}_{\mathrm{conv}}(c_t), \qquad a = \sigma(\theta) \in (0,1)^C$$
$$h_t = (\mathbf{1} - a) \odot \tilde{x}_t + a \odot \tilde{c}_t, \tag{5}$$

where $x_t \in \mathbb{R}^C$ denotes the raw input at time t, $c_t \in \mathbb{R}^C$ denotes the causal convolution output, $\mathrm{LN}_{\mathrm{raw}}$ and $\mathrm{LN}_{\mathrm{conv}}$ are layer-normalization operators, σ is the logistic sigmoid, and $\odot$ denotes element-wise multiplication. Because the weights $1 - a_i$ and a_i are non-negative and sum to one for every channel i, the operation constitutes a convex combination of the two normalized branches. This makes the gate interpretable: values near zero indicate that the feature is treated as static, while values close to one emphasize short-term temporal context. Initialization places a near the raw branch, so the network initially relies predominantly on raw features and incorporates temporal context only when beneficial.

Regularization explains how unused gates are suppressed in practice. The parameters that generate a are subjected to weight decay, which penalizes departures from small magnitudes. When a channel's temporal branch does not improve the supervised objective, the pressure from the loss, combined with weight decay, drives its gate back toward zero, effectively closing that temporal path. Conversely, if short-term dynamics provide predictive benefit, the corresponding gate grows during training and remains open. This yields an adaptive, interpretable allocation of temporal context across channels without forcing rigid prior decisions.

This design was preferred over deterministic feature selection driven by the autocorrelation diagnostics presented in Section 2.3.1. While the ACF provides a valuable indication of temporal dependence, it can be misleading when dependence is weak but predictive, or when autocorrelation does not translate into improved prediction under supervision. In this study, the ACF analysis was employed solely to justify the inclusion of a learnable gate, rather than as a rule for how and where to apply temporal encoding. Allowing the gate to be learned under the anomaly classification objective prevents information loss from premature feature exclusion, captures interactions that simple correlation cannot express, and maintains computational efficiency by restricting convolutions to short, easily parallelized contexts.

3.2.3. Trunk initialization

The classifier was constructed using the same backbone as the regression model, with weights transferred from the regressor after it was trained. This initialization strategy enables the classifier to begin with a representation already optimized for the regression task, rather than starting from random parameters. Since the anomaly detection objective is linked to the regression task (anomalies are defined as extreme values of the predicted quality indicator), the learned trunk provides a stable, informative feature space that can be difficult to establish when training on sparse anomalous samples alone.

Formally, let the regression trunk be denoted as $\phi(x, \Theta_{\mathrm{reg}})$ with parameters Θ_{reg} learned from the regression task. The classifier trunk is then initialized by:

$$\phi(x, \Theta_{\mathrm{cls}}) \leftarrow \phi(x, \Theta_{\mathrm{reg}}) \tag{6}$$

where Θ_{cls} are the classifier trunk parameters. This transfer ensures that the classification network begins from a meaningful latent representation rather than unstructured random weights, and its effect is later isolated in the ablation study by comparison against an otherwise identical randomly initialized classifier.

This approach is preferable to training the classifier from scratch, as anomalies are rare and imbalanced. A randomly initialized model could struggle to form useful feature hierarchies under such conditions, leading to unstable training and poor generalization. By contrast, the transferred weights provide a well-structured starting point, thus enabling the classifier to focus on discriminating rare events.

3.2.4. Group-aware decomposition

The embeddings produced by the gated trunk still mix two sources of variation: systematic group offsets and local fluctuations within groups. To reflect the strong grouping effects identified in Section 2.3.2, the classifier explicitly disentangles these components through a mean-pooling operation followed by two separate logit paths.

Consider a row embedding $z_i \in \mathbb{R}^d$ belonging to group $g(i)$. Let $g(i)$ denote the set of all indices of rows that belong to the same group as i, and $|g(i)|$ its size (the number of rows in that group). The group-average embedding is computed as:

$$\bar{z}_{g(i)} = \frac{1}{|g(i)|} \sum_{j \in g(i)} z_j, \tag{7}$$

that is, the mean of all embeddings within the same group. This pooled representation captures the between-group effect and is passed through a two-layer fully connected network to produce a centered group logit $\ell^{\mathrm{group}}_{g(i)}$. To isolate the within-group signal, the group mean is subtracted from each row embedding:

$$z_i^{\mathrm{res}} = z_i - \bar{z}_{g(i)}, \tag{8}$$

yielding a residual embedding that represents how much row i deviates from its group context. These residual embeddings represent the deviations of each row from its group context and are passed through a single linear layer to produce per-row logits $\ell_i^{\mathrm{residual}}$.

The final logit is obtained by summing the group-level and row-level contributions,

$$\ell_i = \ell^{\mathrm{group}}_{g(i)} + \ell_i^{\mathrm{residual}}, \tag{9}$$

so that each prediction reflects both the systematic offset of the group and the local row-level deviation. This design ensures that anomalies are detected not only as absolute deviations from a global baseline, but also as unusual fluctuations relative to the group context, which is essential in non-i.i.d. data.

Compared to allowing a generic deep network to learn group effects implicitly, this explicit decomposition primarily improves stability and generalization. By constraining the group-average path to capture systematic between-group offsets and the residual path to capture within-group deviations, the two sources of variation are disentangled during training. This prevents group-level shifts from overwhelming the row-level signal and reduces variance in the learned decision boundary. Notably, the ICC analysis in Section 2.3.2 revealed that the between-group variance is several times larger than the within-group variance for many features. Without decomposition, this imbalance could dominate the embeddings, suppressing the contribution of subtle within-group fluctuations. By enforcing separation, the model is better aligned with the data's variance structure, leading to more robust anomaly detection under non-i.i.d. conditions.

At this stage, the classifier produces an uncalibrated logit for every row. By uncalibrated, we mean that although a sigmoid transformation could map these logits into the interval $[0,1]$, the resulting values cannot yet be interpreted as accurate probabilities. For example, a predicted logit of 0.9 under the sigmoid function may, in practice, correspond to an anomaly frequency closer to 0.6. NNs are typically well ranked but poorly calibrated, which motivates a dedicated calibration stage. The

subsequent section introduces calibration and fusion, in which these logits are converted to calibrated probabilities, combined with the regressor predictions, and thresholded to yield the final combined anomaly decisions.

3.3. Calibration and fusion

While the classifier produces logits that separate normal from anomalous rows, the regression task also provides valuable information for anomaly detection. In particular, the residual-boosted regressor outputs continuous predictions of the target variable, which can be compared to the anomaly threshold τ_{QI}, which for the current use case is defined in Section 2.2.2. To quantify this relation, a margin is computed for each row as the signed distance between the predicted value and the threshold:

$$m_i = \tau_{QI} - \hat{y}_i, \tag{10}$$

such that large positive margins indicate rows safely above the defect threshold, while large negative margins signal a substantial likelihood of an anomaly. Margins close to zero correspond to borderline cases where classification is most uncertain. This continuous score complements the classifier logits by providing a physically interpretable measure of how close each prediction lies to the anomaly bound, thereby providing a stable signal even when classification models fail to capture rare dips.

Then, the anomaly detector logits ℓ_i are converted into calibrated probabilities using Platt scaling [51]. A logistic regression is fitted on the validation set,

$$p_{AD,i} = \sigma(a * \ell_i + b), \tag{11}$$

with parameters a, b chosen to minimize cross-entropy against ground-truth anomaly labels. This step corrects for neural classifiers' tendency to be poorly calibrated, ensuring that predicted values reflect empirical anomaly frequencies.

The calibrated anomaly probability is then transformed back into log-odds,

$$s_i = \text{logit}(p_{AD,i}) = \log \frac{p_{AD,i}}{1 - p_{AD,i}}, \tag{12}$$

and combined with the regression margin in a stacked logistic regression:

$$\text{logit}(p_{FUSED,i}) = \beta_0 + \beta_1 * s_i + \beta_2 * m_i, \tag{13}$$

optionally extended with an interaction term $\beta_3 s_i m_i$. This second-stage fusion model is trained on the validation set and outputs the fused probability

$$p_{FUSED,i} = \frac{1}{1 + \exp(-\beta_0 - \beta_1 * s_i - \beta_2 * m_i)}. \tag{14}$$

The fused probability $p_{FUSED,i}$ is finally thresholded at τ_{anom}, selected on the validation set to maximize the F_1 score. This threshold is held fixed during test evaluation.

Through this procedure, classifier confidence and regression distance-to-threshold are integrated into a single calibrated probability estimate. This design leverages the classifier's ranking ability, the regressor's physical interpretability, and the stability of logistic calibration, yielding robust anomaly detection under non-i.i.d. conditions.

3.4. Training protocol

The proposed framework is trained in sequential stages, beginning with the boosted-tree baseline and resulting in calibrated anomaly probabilities fused with regression margins. Each stage is designed to address a specific limitation of the raw data or model outputs, and together they yield a robust dual-task pipeline.

The first stage trains the GBDT baseline directly on the raw feature space. The model is cross-fitted, such that each sample is predicted only by trees that did not see it during training, thereby producing unbiased OOF predictions. These OOF predictions form the foundation for the residual modeling stage while avoiding leakage and optimistic bias.

In the second stage, residuals are computed as the difference between the actual target values and the OOF predictions. These residuals represent systematic errors of the boosted ensemble that a more flexible model can correct.

The third stage trains the residual MLP to model these residuals. Before training, the features are standardized using statistics computed on the training partition, ensuring consistent scaling across partitions. The MLP is trained with MSE loss, yielding a trunk of hidden layers that learns smooth, high-order corrections to the boosted-tree errors.

In the fourth stage, the trained trunk from the residual MLP is transferred to initialize the anomaly detector. Then, the fifth stage trains the anomaly detector on standardized features using focal binary cross-entropy to handle the extreme class imbalance between normal rows and rare anomalies. A critical component here is the *group-aware batch sampler*, introduced to stabilize the training dynamics under non-i.i.d. conditions. Each mini-batch is constructed by sampling a number of rows from multiple groups (coils), rather than filling the batch with random rows. This ensures that both within- and between-group variation is represented in every update, preventing long coils from dominating the gradient signal. Moreover, by balancing the representation of coils, the sampler ensures that group-average embeddings are estimated from sufficient context, which is essential for the group-aware decomposition to function correctly. Without such balanced batches, group means would be noisy or underrepresented, undermining the stability of the entire framework. Training is further stabilized with gradient clipping and a warm-up, along with a cosine learning rate schedule.

In the sixth stage, the classifier logits are calibrated with Platt scaling on the validation set. In the seventh stage, these calibrated anomaly probabilities are fused with the regression margins (Section 3.3). Finally, in the eighth stage, a decision threshold τ_{anom} is selected on the validation set. The threshold can, in principle, be tuned to the application context: fault-critical settings may prioritize recall to minimize false negatives, while more tolerant settings may prioritize precision. In this study, τ_{anom} is selected to maximize the F_1 score, the harmonic mean of precision and recall, to balance the two error types.

After completing these steps, the model is evaluated on the held-out test partition of the current CV fold, using the calibration parameters, fusion model, and threshold selected on validation. This procedure is repeated for all K outer folds, ensuring that each group is used at least once for testing and that the reported metrics reflect a robust, cross-validated evaluation of the framework. A complete step-by-step summary of the training sequence is provided in Algorithm 1 in Appendix B.

The proposed framework thereby provides a solution for predicting continuous quality indicators and rare anomalies from a grouped, temporally correlated dataset.

4. Experimental setup

To ensure fair and reproducible evaluation, all experiments were conducted under a common training and validation protocol. Two sets of baselines are considered. First, the existing physics-based predictor deployed at the plant is reported as a reference for CT prediction accuracy. Second, controlled ablation and baseline configurations are constructed to quantify the contribution of each architectural component of the proposed framework. For the regression task, these include comparisons between the proposed regressor, raw GBDT predictions, and a residual-boosted regressor with temporal encoding. For anomaly detection, the weight transfer, temporal encoder, causal gate, and group-aware decomposition are stepwise removed from the anomaly detector. Together, these baselines establish the foundation against which the whole framework is assessed.

4.1. Physics-based baseline

The physics-based model currently employed by the company for ROT cooling control is a first-principles approach that accurately models strip cooling and temperature evolution from finishing to coiling. It is implemented as a Model Predictive Controller (MPC) that predicts the required cooling intensity and duration to meet specific temperature profiles critical to product quality.

The framework, at its core, blends first-principles heat-transfer equations with empirically calibrated coefficients to predict the strip temperature from finishing to coiling. Every strip segment is divided into surface header zones (jet impingement and settling regions) and through-thickness computational cells. At each time step, an energy balance combines radiative, convective, and water-cooling heat fluxes with internal conduction and latent heat from phase transformations. Water-cooling and water-settling heat transfer coefficients are stored in proprietary look-up tables indexed by surface temperature. Four anchor points per zone are linearly interpolated. These tables, as well as emissivity and convection factors, were derived from in-house test campaigns. The model assumes constant strip speed, density, and specific heat within each zone, uniform air properties for convection, one-dimensional heat conduction through thickness, and negligible water temperature rise. Radiative exchange uses a grey-body emissivity and a constant view-factor multiplier to account for roll reflections.

4.2. Baseline and comparison configurations

To thoroughly evaluate the proposed framework, both the regressor and anomaly detector models are contrasted against a series of baselines. The regressor is evaluated against the physics-based model, the raw GBDT predictions, and a residual-boosted NN with temporal encoding. To isolate the contribution of each component in the proposed anomaly detector, a series of ablation studies is conducted to quantify the role of each architectural and training-design element in the anomaly detector. The classifier ablations are therefore divided into structural removals, which alter the detector architecture, and an initialization ablation, which keeps the full architecture fixed while changing only the starting point of the shared trunk.

- **B0 - Physics-based baseline.** The analytical ROT model described in Section 4.1 is reported as the industrial benchmark. This establishes the minimum performance that any data-driven alternative must exceed to be considered competitive.
- **B1 - Raw GBDT predictions.** The GBDT model trained on raw features provides a strong tabular-data baseline. Comparing the hybrid model against B1 demonstrates whether the residual correction provides systematic improvements over a standalone boosted ensemble.
- **B2 - Propsed regressor.** This is the proposed regressor architecture with the residual-boosted MLP as introduced in Section 3.1.
- **B3 - Residual-boosted NN with temporal encoding.** In this configuration, the residual MLP is replaced by a NN with temporal encoding that follows the same compact temporal encoding principle used in the anomaly detector. It is introduced to examine whether short-range dependencies can provide additional gains in continuous CT prediction.
- **C1 - Full anomaly detector.** The complete classifier includes causal temporal convolutions, causal gate fusion, and group-aware decomposition. This configuration represents the proposed anomaly detector architecture in Section 3.2 (without regressor fusion) and serves as the reference point against which all subsequent classifier ablations are evaluated.
- **C2 - No group-aware decomposition.** The trunk output is passed directly to a single logit head, omitting the separation of group-level and within-group contributions. This ablation tests the hypothesis that explicitly disentangling variance sources stabilizes training and improves the detection of within-coil dips. Performance degradation in this setting would confirm the need to model a grouped, non-i.i.d. structure explicitly.
- **C3 - No temporal encoder.** In this configuration, input features are passed directly to the trunk without convolutional context. The aim is to test whether the short-range dependencies identified in Section 2.3.1 are critical for detecting transient dips, or whether anomalies can be detected without explicitly encoding temporal history.
- **C4 - Fully-Connected baseline.** In this configuration, the anomaly detector is reduced to a plain fully-connected network (FCN) trained with focal binary cross-entropy loss, without temporal convolution or group-aware decomposition. Each row is treated independently of its temporal neighbors and of its coil identity, so the model cannot exploit either short-term dependencies or between-group or within-group variance. This ablation serves as a minimal deep-learning baseline, testing whether the performance gains of the proposed framework arise from the architectural innovations or simply from the nonlinearity of a neural classifier.
- **C5 - Random-initialized full classifier.** This configuration keeps the full anomaly detector architecture unchanged, including the temporal encoder, learnable gated fusion, and group-aware decomposition, but replaces regression-based trunk initialization with random initialization. It is introduced to isolate the contribution of weight transfer independently from the architectural contributions.

All classifier ablation variants are trained with the same group-aware sampler, optimization procedure, and calibration/fusion steps as the whole model to ensure comparability. In addition, because temporal encoding requires valid causal windows, the first rows of each coil are excluded, and all reported models are therefore evaluated on the same common subset. The selected baselines are intended to cover the three most relevant comparison axes for this problem: the industrial process benchmark, a strong tabular-data learner, and simplified deep-learning variants that isolate whether performance gains arise from temporal and group-aware structure rather than from neural nonlinearity alone. Generic unsupervised anomaly-detection methods were not used as primary baselines because the present task is formulated as supervised threshold-based rare-event detection under grouped, leakage-safe evaluation, rather than as unlabeled anomaly discovery. All of the baselines and ablation configurations are summed up in Table 2.

4.3. Evaluation metrics

The performance of the models is assessed on two complementary levels: continuous prediction of the quality indicator, and binary detection of rare anomalies.

For the regression task, the root mean square error (RMSE) is used as the primary measure, since it penalizes larger deviations and provides a robust measure of overall accuracy. In parallel, the coefficient of determination (R^2) is reported to quantify the proportion of variance in the target variable explained by the model, providing a complementary indicator of goodness of fit. However, because both RMSE and R^2 summarize performance across all samples, they remain insensitive to rare but critical dips. A regressor may achieve low RMSE and high R^2 by fitting the majority of normal cases while failing to capture infrequent but process-critical anomalies.

To evaluate anomaly detection performance, models are assessed according to their architecture:

- **Regressor-only anomaly detection.** Predictions are converted to binary labels by comparing the continuous output directly to the process threshold:

$$\hat{s}_i = \mathbf{1}\,[\hat{y}_i \le \tau_{QI}], \tag{15}$$

where $\hat{y}_i$ is the predicted quality indicator value, and τ_{QI} is the lower quality bound defined by the steel grade recipe. This is how a single

Table 2
Baselines and ablation configurations used in the study. B0-B3 denote regression baselines; C1-C5 denote classifier variants.

Code	Configuration	Description
B0	Physics-based baseline	Analytical ROT model currently deployed for cooling control. Provides the industrial benchmark for CT prediction.
B1	Raw GBDT predictions	GBDT trained directly on raw features. Strong tabular-data baseline; benchmark for proposed residual-boosted regressor.
B2	Proposed Regressor	GBDT residual-boosted MLP. The full proposed regressor architecture.
B3	Residual-boosted NN with temporal encoding	GBDT residual-boosted NN with temporal encoding. Evaluation of the temporal encoding in the regression task.
C1	Full anomaly detector - without fusion	Proposed classifier including causal temporal convolutions, causal gate fusion, and group-aware decomposition. Serves as the reference model.
C2	No group-aware decomposition	Trunk embeddings are passed to a single logit head. Evaluates the effect of explicitly separating group-level and within-group contributions.
C3	No temporal encoder	Input features bypass the convolutional context. Tests the importance of explicitly modeling short-range dependencies for detecting anomalies.
C4	Fully-Connected Network only	No temporal or group-aware structure - Deep learning baseline.
C5	Full anomaly detector - random initialization	Same with C1 with random intialization (no weight transfer)

regressor can be utilized to detect anomalies: any prediction below the bound is considered anomalous.

- **Classifier-only anomaly detection.** The anomaly detector produces logits that are Platt-calibrated into probabilities $p_{AD,i}$. Binarized labels are obtained by thresholding these probabilities at τ_{anom}:

$$\hat{s}_i = \mathbf{1}\,[p_{AD,i} \geq \tau_{anom}]. \tag{16}$$

The threshold τ_{anom} is chosen on the validation set to maximize the F_1 score, balancing recall and precision. In more fault-critical contexts, τ_{anom} can instead be tuned to maximize F_β with $\beta = 2$, thereby placing greater emphasis on recall.

- **Fused model anomaly detection.** For the full framework, anomaly predictions follow the fusion procedure of Section 3.3. The calibrated classifier probabilities (converted to log-odds) and regression margins are combined in a logistic regression, yielding fused probabilities $p_{FUSED,i}$. Binary labels are then defined as

$$\hat{s}_i = \mathbf{1}\,[p_{FUSED,i} \geq \tau_{anom}], \tag{17}$$

with τ_{anom} again selected on the validation set to maximize F_1.

For all three cases, the ground-truth labels are defined as $s_i = 1$ whenever the measured target value falls below τ_{QI}. The following metrics of accuracy, recall, precision, F_1, and F_β are reported for the anomaly class.

Accuracy measures the overall proportion of correct predictions, Recall quantifies the ability to detect rare anomalies, Precision reflects the reliability of positive predictions, F_1 balances the two, and F_β generalizes the trade-off by weighting recall β times more heavily than precision.

All models are trained separately for each outer CV fold. Within each fold, the training set is split into an inner training set and a validation set. The model is fit on the inner training set, monitored on the validation set, and the checkpoint with the lowest validation loss is retained. The validation set is then used for calibration, fusion, and threshold selection. Final metrics are averaged for the held-out test partition of each fold, ensuring unbiased evaluation.

4.4. Implementation and computational setup

All implementation details, the computational setup, and the hyperparameter values are presented in Appenidx C.

5. Results and discussion

In the following section, the performance of the quality indicator regression and anomaly detection tasks is reported separately. Afterwards, the results of the ablation study are presented to showcase the contributions of the temporal and group-aware parts of the framework to overall performance. Finally, case-study-specific results and joint anomaly-detection and regression plots are presented to illustrate the framework's efficacy in realistic production scenarios.

Table 3
Regression performance on the common test subset used for the temporal comparison for the entire CV split. Best values in **bold**; lower is better for RMSE, higher is better for R2 score.

Model	RMSE (μ)	RMSE (σ)	R2 (μ)	R2 (σ)
Physics-based model (B0)	0.0293	0.00115	−1.664	0.19648
GBDT-only (B1)	0.0131	0.00029	0.4649	**0.01094**
Proposed Regressor (B2)	0.0128	**0.00032**	0.4927	0.01265
GBDT-Temporal Encoding (B3)	**0.0127**	0.00042	**0.4975**	0.02451

5.1. CT profile regression results

Table 3 reports the regression performance of the four models under comparison on the test subsets averaged across all of the CV folds, while Fig. 12 shows the corresponding predicted-true scatter plots for the first CV fold, also referred to as predicted vs true scatter plots. The plot of the physics-based model currently used in the plant (B0) is shown in Fig. 12a. With an ideal regressor, all of the predicted values would equal the true values, and thus all of the points in the plot would lie on the $y = x$ line. In the plot of the physics-based baseline, we can see that the scatter points do not lie on the $y = x$ line and show no linear trend with it; therefore, the regressor predicts erroneous values. This translates to an average RMSE of 0.0293 and a negative R^2, confirming that it fails to capture the data's variability and can be outperformed even by a constant-mean predictor. This underperformance can be attributed to the simplifying assumptions embedded in the first-principles formulation - constant strip speed and material properties, one-dimensional conduction, and fixed heat-transfer coefficients. These approximations are particularly problematic for the heavy-gauge steel grade considered in this study, where complex through-thickness conduction, phase transformation effects, and water-impingement dynamics significantly influence cooling behavior. As shown in the gauge distribution of the dataset (Fig. 5a), the prevalence of thick-gauge coils exacerbates these discrepancies, amplifying the model's inability to estimate the CT profiles.

The raw GBDT baseline (B1) substantially improves accuracy, reducing the RMSE by more than 50% and achieving a mean R^2 of 0.47. This confirms that tabular gradient boosting is a strong learner for this task, capturing nonlinear interactions between the process parameters and the quality indicator. However, the predictions tend to remain close to the mean, failing to reproduce very low or very high values accurately. This regression-to-the-mean behavior makes it challenging for the GBDT alone to capture rare, extreme deviations, which are crucial for anomaly detection.

The proposed residual-boosted regressor achieves the best performance, with the lowest mean RMSE (0.0128) and an improved mean R^2 (0.49). The improvements, although incremental compared to the GBDT, are consistent across CV folds as indicated by the low variance of both RMSE and R^2. Significantly, the addition of the neural

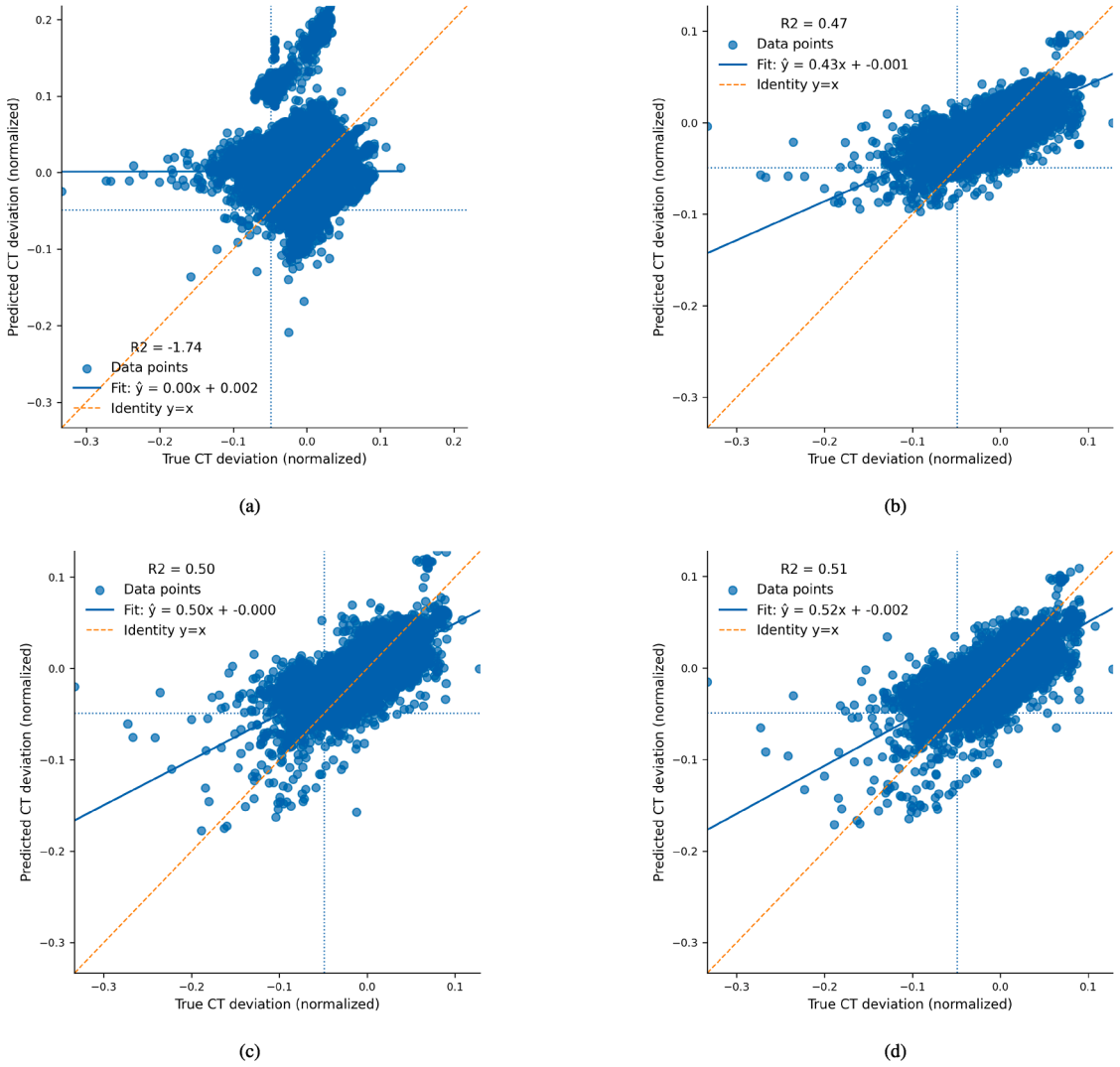

Fig. 12. Predicted versus true coiling temperature (CT) deviation on the first CV fold of the test data. (a) The physics-based baseline (B0) shows large systematic errors and a negative R^2. (b) Raw GBDT baseline (B1) improves fit but exhibits regression-to-the-mean behaviour, underestimating both very cold and very hot deviations. (c) Proposed residual-boosted regressor (B2) achieves better alignment with the identity line (oranged dashed line), especially in the cold regime where anomalies are concentrated, reflecting improved generalization at the extremes, (d) GBDT-Temporal residual NN achives only marginal improvement.

correction shifts more predictions closer to the identity line, particularly at the extremes. In the high-value range, the model begins to capture values previously underestimated, while in the low-value range - where anomalies are of most significant concern for the presented use case - the scatter plot in Fig. 12c shows a marked improvement, with more points aligned along the $y = x$ line compared to the GBDT (Fig. 12b).

Adding temporal encoding to the residual NN produces only a marginal improvement over the proposed residual-boosted regressor, with a slightly lower mean RMSE (0.0127) and a slightly higher mean R^2 (0.4975) (Fig. 12d). Although this indicates that the short-range temporal structure identified in Section 2.3.1 can also be exploited on the regression side, the magnitude of the gain remains very small and is comparable to the fold-to-fold variability observed across the CV splits. This suggests that most of the predictable continuous structure is already captured by the GBDT baseline, while the residual learner operates on a reduced error signal in which the remaining temporal information is comparatively weak. In this setting, the temporal encoder yields a numerically positive but practically limited refinement, and therefore does not justify replacing the simpler residual MLP as the main regression model.

Taken together, these results demonstrate that the proposed residual-boosted design successfully transfers information from the boosted tree baseline to the NN, improving generalization without overfitting. Compared to the physics-based model currently deployed in production, the proposed regressor offers a significant gain in predictive accuracy, providing a stronger foundation for downstream anomaly detection, while the comparison with the additional temporal encoding shows that no substantial improvement can be obtained.

5.2. Anomaly detection performance

Fig. 13 presents the confusion matrices for the four anomaly detection strategies: the physics-based baseline (B0), the residual-boosted regressor with thresholding at τ_{QI}, the classifier-only model with Platt-calibrated probabilities, and the fused model combining regression margins with calibrated classifier probabilities. Table 4 summarizes the anomaly detection metrics for the anomaly class (denoted by the + next to the metric), and Fig. 14 shows the corresponding precision-recall curves with operating points annotated at the validation-selected F_1 threshold.

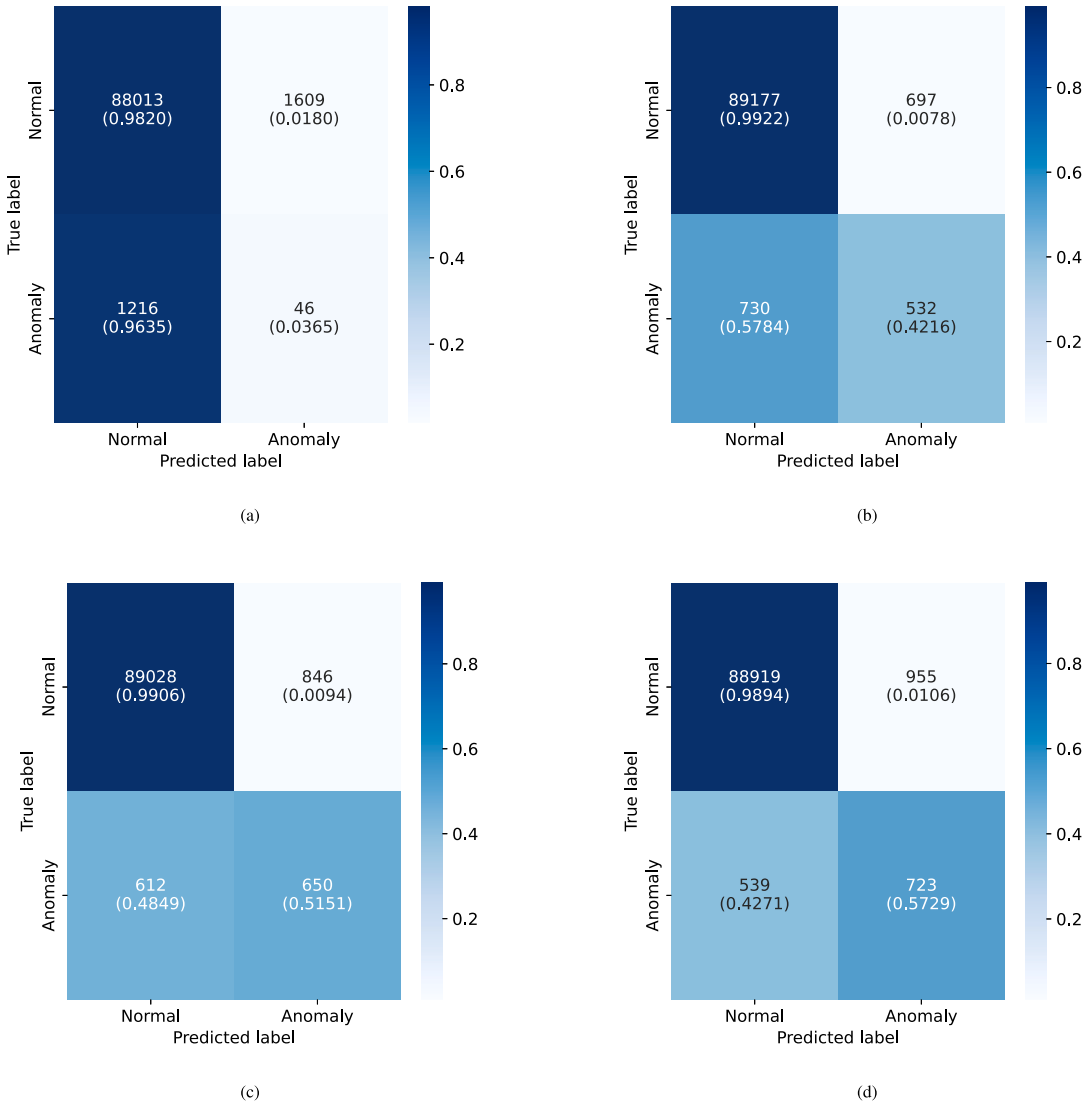

Fig. 13. Confusion matrices for four anomaly detection strategies on the test data of the first CV split: (a) physics-based baseline (B0), (b) residual-boosted regressor with anomalies flagged when predictions fall below τ_{QI}, (c) classifier-only model with Platt-calibrated probabilities thresholded at τ_{anom} selected for F_1, and (d) fused model combining regression margins with calibrated classifier probabilities. Each cell shows both the raw counts and the normalized proportions. For the classifier-only and fused configurations, the total counts are lower than for the regressor-based approaches because the first three time steps of each coil are dropped when constructing input windows.

Table 4
Anomaly detection performance on the test data averaged across all of the CV splits. All of the values are reported as mean ± standard deviation. Best values in **bold**; higher is better for all metrics.

Model	F_1 +	F_2 +	Precision +	Recall +
Physics-based baseline (B0)	0.04	0.04	0.04	0.03
Regressor only	0.392 ± 0.0515	0.321 ± 0.0496	0.632 ± 0.0354	0.286 ± 0.0467
Classifier only (C1)	0.4657 ± 0.0317	0.4682 ± 0.0391	0.4653 ± 0.0452	0.4708 ± 0.0483
Proposed-Fused	**0.4846 ± 0.0334**	**0.4852 ± 0.0391**	**0.4849 ± 0.0314**	**0.4858 ± 0.0442**

The physics-based baseline (B0, Fig. 13a) fails to identify anomalies, with recall close to zero and overall precision and F_1 below 0.05 (Table 4). This outcome reflects the same systematic underestimation observed in regression, leaving rare dips in the quality indicator entirely undetected.

Thresholding the residual-boosted regressor at τ_{QI} (Fig. 13b) results in improved detection performance, yielding $F_1 = 0.39$ and $F_2 = 0.32$. While precision is high (0.632), the detection capability remains limited (as justified by the reduced recall at 0.29) because the regressor, while accurate in predicting mean quality indicator behaviour, does not produce sufficiently sharp responses at the boundaries. As seen in the scatter plots of Section 5.1, the GBDT + NN hybrid reduces error. Still, it exhibits regression-to-the-mean behaviour, making it ill-suited as a standalone anomaly detector. This justifies the addition of a dedicated classifier to model and capture rare anomalies explicitly.

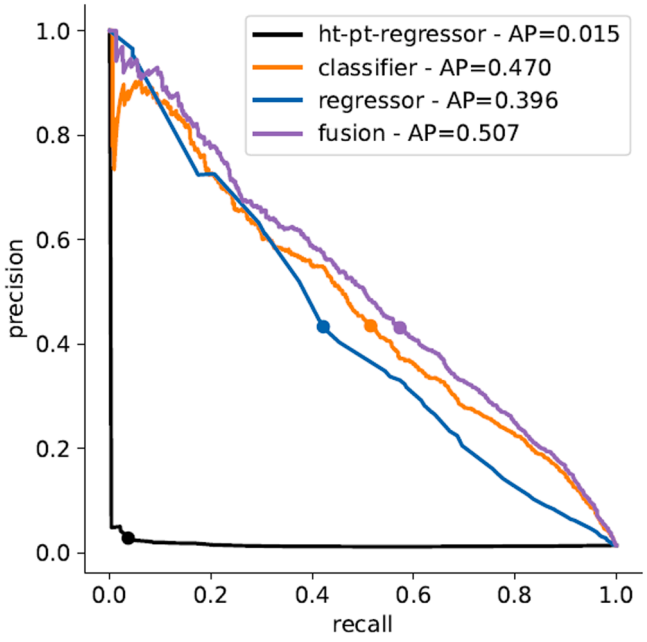

Fig. 14. Precision-recall curves for the four anomaly detection strategies: physics-based baseline (B0), regressor-only thresholding at τ_{QI}, classifier-only with calibrated probabilities, and fused model with logistic combination of margin and probability. The annotated dot marks the operating point chosen on validation to maximize F_1. The average precision (AP) score for each model is reported in the legend. The fused model achieves the highest performance, evident by the largest area under the curve and the highest AP. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Introducing the coil-aware anomaly detector (Fig. 13c) markedly increases recall and yields balanced precision-recall trade-offs, with F_1 and F_2 both around 0.47 (Table 4). The classifier leverages temporal encoding and group-aware decomposition, overcoming the regressor's averaging tendency and improving sensitivity to localized low values in the quality indicator profiles.

Finally, the fused model (Fig. 13d) achieves the best overall performance. By combining the classifier-calibrated probability with the regression margin using a logistic fusion scheme (Section 3.3), the model achieves F_1 and F_2 scores of around 0.485, outperforming all other configurations. The precision-recall (PR) curves in Fig. 14 confirm that the full model improves the precision-recall envelope across thresholds, with the operating point tuned for F_1 (annotated dot) lying on the upper frontier of all curves. This is also evident from the higher average precision (AP) score reported in the plot legend.

Taken together, these results highlight two key findings: (i) regression accuracy alone does not guarantee anomaly detection performance, due to regression-to-the-mean effects at the extremes, and (ii) combining a dedicated anomaly classifier with regression margins through calibrated fusion provides the most accurate and balanced detection of rare anomalies.

This behaviour can be explained by the complementary error profiles of the two components. As reported in Table 4, the regressor-only approach achieves relatively high precision but low recall, detecting anomalies conservatively and often missing low-value dips. In contrast, the classifier achieves higher recall by exploiting temporal and group-aware context, but at the expense of false positives. Logistic regression fusion leverages the regressor margin as a calibrated auxiliary feature, weighting the classifier output against the regressor's precision. The result is a fused probability that inherits the classifier's recall strength while mitigating spurious detections, effectively balancing sensitivity and specificity.

It is worth noting that the absolute performance levels in the presented use case remain modest. Precision and recall are balanced but not high in absolute terms, which is partly due to the nature of the use-case data. First, the CT is only measured at the coil-segment level, rather than at the finer resolution of spray zones or local cooling dynamics, limiting the granularity of the available signal. Second, the prediction task is particularly challenging because it attempts to infer the post-cooling temperature outcome using only setpoints and upstream process parameters, without direct measurements from the cooling process itself. This design choice, however, is deliberate: by predicting the quality indicator and flagging anomaly risk before the cooling process occurs, the framework provides lead time for preventive adjustment of cooling settings. Finally, the limited number of confirmed fault samples further constrains achievable performance. The rarity of the anomalies biases the learning process toward the dominant healthy regime, making it challenging to capture the full variability of fault signatures and limiting F_1 scores even in architectures explicitly designed to handle

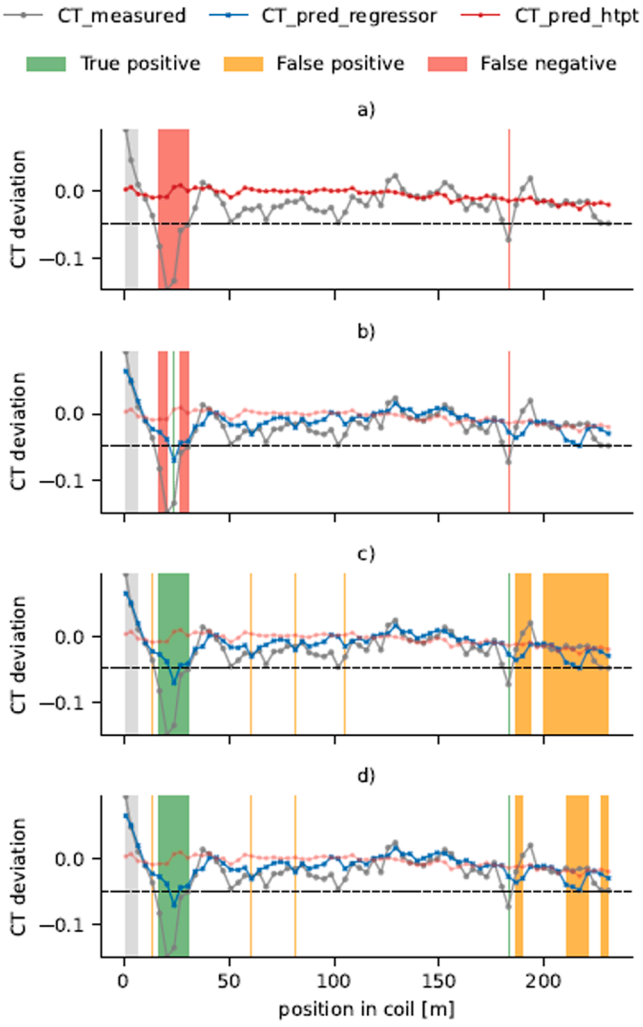

Fig. 15. Case study of a coil with multiple cold dips. Panels show anomaly detection overlays for (a) physics-based baseline (B0), (b) residual-boosted regressor thresholded at τ_{QI} (c) classifier-only with calibrated probabilities, and (d) fused model. Grey = measured CT profile, red = physics-based predictions, blue = regressor predictions. Shaded areas mark true positives (green), false positives (orange), and false negatives (red). The regressor begins to capture some dips but misses many, the classifier recovers most anomalies at the cost of false positives, and the fused model reduces spurious detections while preserving recall. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

imbalance [52–54]. In practice, the proposed approach is intended as a data-driven replacement for the existing physics-based predictor, with the added functionality of anomaly risk indication based on CT dips that are otherwise disregarded as noise in the literature, offering a decision-support signal that complements the existing control logic. Even when absolute anomaly metrics remain modest, the reduction of false positives achieved by temporal context, group-aware decomposition, and calibrated fusion is operationally important because unnecessary interventions are costly and may degrade throughput.

5.3. Ablation study

Table 5 reports both structural ablations and an additional initialization ablation for the anomaly detector. Overall, C1 remains the best-balanced configuration, achieving the strongest combined F_1 and F_2 performance while maintaining closely matched precision and recall. This indicates that the joint use of temporal encoding, gated fusion, group-aware decomposition, and transferred initialization provides the most stable detection of rare CT dips.

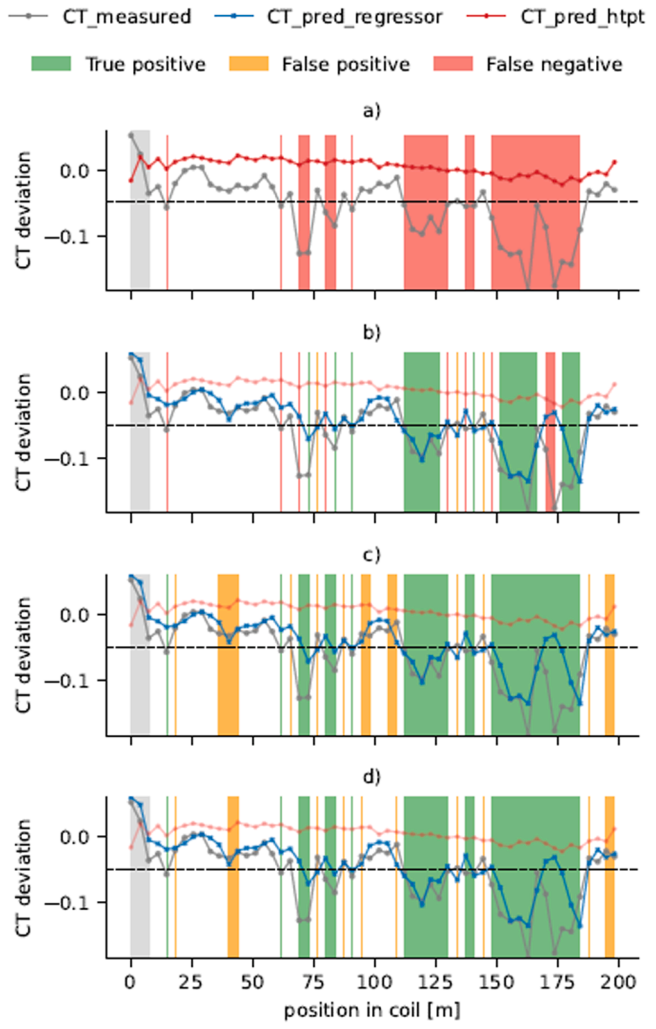

Fig. 16. Case study of a more challenging coil with weaker and sporadic dips. Coloring and layout as in Fig. 15. The regressor struggles in this case, maintaining high precision but low recall. The classifier captures all anomalies but produces numerous false alarms, whereas the fused model balances the two, reducing false positives while retaining all correct detections. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 5
Ablation study on the proposed classifier. C1 is the full anomaly classifier without probability fusion; C2 removes the group-aware decomposition; C3 removes temporal encoding; and C4 reduces the detector to a plain fully connected network trained with focal BCE. C5 keeps the full classifier architecture fixed but replaces transferred trunk initialization with random initialization. C1 values are repeated from Table 4 for completeness. Metrics are computed from test data averaged across all CV splits. All of the values are reported as mean ± standard deviation.

Model	F_1 +	F_2 +	Precision +	Recall +
C1	**0.4657 ± 0.0317**	**0.4682 ± 0.0391**	**0.4653 ± 0.0452**	0.4708 ± 0.0483
C2	0.4526 ± 0.0471	0.4637 ± 0.0575	0.4377 ± 0.0397	0.4722 ± 0.0660
C3	0.4359 ± 0.0406	0.4567 ± 0.0609	0.4085 ± 0.0233	0.4731 ± 0.0783
C4	0.4348 ± 0.0343	0.4604 ± 0.0350	0.400 ± 0.0477	**0.4800 ± 0.0407**
C5	0.4632 ± 0.0407	0.4672 ± 0.0498	0.4596 ± 0.0427	0.4706 ± 0.0585

When the group-aware decomposition is removed (C2), performance degrades moderately, with the main loss appearing in precision rather than recall. Precision decreases from 0.465 to 0.438, whereas recall remains at approximately 0.472. This is consistent with the strong between-coil variance identified in Section 2.3.2. Without explicitly separating coil-level offsets from within-coil deviations, the detector produces more false positives in coils with shifted baseline behaviour.

Omitting temporal encoding (C3) leads to a stronger degradation. Although recall remains nearly unchanged at 0.473, precision drops further to 0.409, with corresponding reductions in both F_1 and F_2. This indicates that a short causal context is important for suppressing noisy detections and for linking successive rows into coherent local anomalies. Without temporal encoding, the detector remains sensitive to anomalies, but it becomes less accurate.

The plain fully connected baseline (C4) shows the same trade-off more clearly. It attains the highest recall, 0.480, but the lowest precision, 0.400, and consequently underperforms the structured variants in both F_1 and F_2. This behaviour suggests an over-sensitive detector that captures many anomalous rows but also misclassifies a substantial number of normal observations, confirming that, for the correct identification of anomalies, both temporal and group-aware structures are important.

Finally, the random-initialized full classifier (C5) remains competitive, achieving $F_1 = 0.463$ and $F_2 = 0.467$, but it is consistently slightly inferior to proposed architecture C1 with weight transfer initialization, which reaches $F_1 = 0.465$ and $F_2 = 0.468$. The main difference again appears in precision, which decreases from 0.465 to 0.460, while recall remains essentially unchanged at approximately 0.471. This indicates that transferred initialization modestly but consistently improves discrimination once the full detector architecture is present. At the same time, the larger drops observed in C2-C4 indicate that the dominant gains arise from explicitly modeling short-range temporal context and group-level structure, whereas weight transfer provides additional refinement rather than being the primary source of performance.

It is important to note that C1 corresponds to the standalone classifier before probability fusion. As shown in Table 4, the logistic fusion of classifier probabilities with regressor margins further improves recall beyond the FCN baseline, while simultaneously restoring precision. This highlights the complementary role of fusion in balancing sensitivity and specificity.

5.4. Case studies - CT profile with anomaly overlays

To complement the aggregate performance metrics and illustrate the application of the proposed framework in a real production scenario, Figs. 15 and 16 illustrate two representative coil profiles with anomaly detection overlays using four different strategies: (a) physics-based baseline (B0) and (b) residual-boosted regressor thresholded at τ_{QI}, (c) classifier-only (C1), and (d) fused model thresholded at the tuned τ_{anom}.

The first example (Fig. 15) demonstrates a rare case of a highly faulty coil with multiple pronounced cold dips. The physics-based baseline fails, as annotated with the red-shaded areas and lines (False Negatives), to capture any anomalies, consistent with its near-zero recall reported in Section 5.2. The regressor-only model begins to identify several dips correctly (True Positives annotated with green-shaded areas and lines). Yet, many remain undetected due to its regression-to-the-mean tendency. The classifier substantially increases recall, detecting nearly all anomalies, but introduces a considerable number of false positives (annotated with orange-shaded areas and lines). The fused model mitigates these spurious detections, as evident from the reduced orange areas, while maintaining high recall, resulting in the best overall alignment between predicted and measured anomalies. This behavior reflects the complementary error profiles described in Section 5.2: the regressor's high precision constrains the classifier's broader sensitivity through the logistic margin fusion.

The second example (Fig. 16) highlights a more challenging case in which anomaly signals are weak and sporadic, as is usually observed in well-behaved coils. Here, all methods struggle, but the progression remains consistent. The regressor maintains high precision with sparse detections, the classifier manages to correctly capture the two abrupt dips but introduces more false alarms, and the fused model balances the two, retaining the correct detections while reducing false positives relative to the classifier alone. This aligns with the precision-recall tradeoffs shown in Table 4, confirming that the fusion inherits the strengths of both components.

Together, these case studies reinforce the quantitative findings: while regressors are effective for stable CT prediction, anomaly detection requires a classifier. Fusion of the two yields the most robust identification of rare CT dips, as evidenced by both overall metrics and coil-level examples.

6. Conclusions

The present study proposes a general framework for predicting quality indicators with anomaly detection in datasets that are inherently grouped and non-i.i.d., while exhibiting short-range temporal dependencies. Although CT was used as a representative case, the methodology applies more broadly to industrial batch processes, segmented time-series data, and other domains where group-level offsets coexist with local sequential structure.

The following conclusions were drawn from the study:

- Abrupt coiling temperature (CT) dips below the lower acceptable temperature bound are not stochastic noise but reliable proxies of surface-quality defects, and therefore should be explicitly modeled rather than discarded during preprocessing.
- Production datasets should first be examined using data diagnostics to identify temporal dependencies and grouping effects. When such structures are identified, they must be explicitly represented in both the model architecture and data splitting to increase model validity and prevent leakage.
- Encoding short-term temporal dependencies and explicitly decomposing between-group and within-group variation improves the model's ability to detect abrupt and localized anomalies in grouped, non-i.i.d. data.
- Combining the mean-seeking tendency of gradient-boosted trees with the flexibility of neural residual modeling improves accuracy at extreme quality indicator values, without harming generalization.
- When anomalies are defined by abrupt dips of the quality indicator, thresholding a conservative regressor's output and fusing it with a higher-recall anomaly detector through logistic regression at the probability level yields a balanced trade-off, reducing false positives with minimal loss of anomaly recall.
- Although performance in the presented use-case is bounded by the granularity and informativeness of available data, and by downstream process influences, the proposed framework improves model efficacy within these constraints, providing more accurate predictions and a quality-risk screening and decision-support layer that can provide earlier warning than the currently deployed predictor.

The end goal is to produce an informative, actionable, and generalizable quality indicator predictor with anomaly detection capabilities for production datasets. To that end, future work should focus on further enhancing the accuracy, interpretability, and robustness of the proposed framework. Physics-informed modeling offers a promising direction, particularly for the showcased use case of CT prediction, where constraining regression margins with simplified energy-balance formulations, embedding spray-zone structure in the temporal encoder, or regularizing latent representations with known thermal-transfer relations could strengthen physical consistency. Another critical avenue is uncertainty quantification, extending the model to provide confidence estimates for both regression margins and anomaly probabilities. This will enable operators to assess alarm reliability and determine the necessary action. Online calibration drift monitoring can ensure robustness under evolving plant conditions. At the same time, limited-label or weakly supervised anomaly learning could help exploit the large volume of available production data where only a small subset of anomalies is confirmed. Finally, the present case study is limited to one HSLA heavy-gauge production regime, so broader applicability across steel grades and product specifications remains to be demonstrated explicitly. Nevertheless, the framework is defined at the methodological level by grouped splitting, short-range temporal encoding, and group-aware variance decomposition, and is therefore not tied to a single recipe. Extending it across grades would require re-evaluating the temporal and grouping diagnostics for the new regime, recalibrating grade-specific CT thresholds and decision boundaries, and potentially conditioning the model on grade or product descriptors. Where shifts in operating windows or group composition are substantial, domain adaptation mechanisms would also be required. Evaluating these extensions across broader HSM production regimes is therefore an important next step toward plant-wide deployment.

Declaration of generative AI in scientific writing

During the preparation of this work, the authors used ChatGPT based on GPT5 in order to improve the readability and language of some parts of the paper. The tool was in no way used to analyze and draw insights from the data, perform literature research, or extract any information other than feedback on the writing style based on the provided inputs. The tool was only used to perform minimal changes and provide feedback based on the provided input text, where the scientific content of the input sentences remains unchanged. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the content of the publication.

CRediT authorship contribution statement

Thanos Kontogiannis: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation; **Dimitrios Zarouchas:** Writing – review & editing, Validation, Supervision, Resources, Methodology, Funding acquisition, Conceptualization; **Nick Eleftheroglou:** Writing – review & editing, Validation, Supervision, Methodology, Conceptualization.

Data availability

The data that has been used is confidential.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Thanos Kontogiannis reports financial support was provided by Dutch Research Council. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research is part of the DEPMAT project (with project number P20-22 / N21022) of the research programme Perspectief, which is partly financed by the Dutch Research Council (NWO). It is also part of the Partnership Program of the Materials innovation institute M2i (https://www.m2i.nl).

The authors gratefully acknowledge Dr. Wanda Melfo, Principal Scientist in the R&D Department of Tata Steel Nederland B.V., for providing the production data, sharing plant evidence linking coiling temperature dips to surface defects, and offering valuable expert insight into the hot rolling process.

Appendix A. ANOVA details and variance components derivation

For observations y_{gi} indexed by group g and instance i, the model is:

$$y_{gi} = \mu + \alpha_g + \varepsilon_{gi}, \quad \alpha_g \overset{iid}{\sim} \mathcal{N}(0, \sigma^2_{\text{between}}), \quad \varepsilon_{gi} \overset{iid}{\sim} \mathcal{N}(0, \sigma^2_{\text{within}}), \tag{A.1}$$

with α_g orthogonal to ε_{gi}. The marginal variance, therefore, decomposes additively as:

$$\text{Var}(x_{gi}) = \sigma^2_{\text{between}} + \sigma^2_{\text{within}}. \tag{A.2}$$

Estimation of these variance components proceeds through ANOVA mean squares. The within-group mean square is:

$$MS_W = \frac{\sum_{g=1}^{G}(n_g - 1)\, s_g^2}{N - G}, \tag{A.3}$$

where s_g^2 is the sample variance inside a group g, n_g is the number of rows in coil g, $N = \sum_g n_g$ is the total number of rows, and G is the total number of groups. Its expectation is therefore is:

$$E[MS_W] = \sigma^2_{\text{within}}. \tag{A.4}$$

Similarly, the between-group mean square and its expectation are defined as:

$$MS_B = \frac{\sum_{g=1}^{G} n_g(\bar{y}_g - \bar{y})^2}{G - 1}, \quad E[MS_B] = \sigma^2_{\text{within}} + \bar{n}\,\sigma^2_{\text{between}}, \tag{A.5}$$

where $\bar{y}_g$ is the mean of y within coil g, $\bar{y}$ is the overall mean, and $\bar{n}$ denotes an effective average group size that corrects for unequal group lengths:

$$\bar{n} = \frac{N - \frac{\sum_{g=1}^{G} n_g^2}{N}}{G - 1}. \tag{A.6}$$

This expression reduces to the common group size n when all groups are equal, but downweights highly imbalanced groups when sizes vary. Incorporating $\bar{n}$ ensures that the estimator of the between-group variance remains unbiased under unequal group sizes [42]. The resulting variance component estimates are

$$\hat{\sigma}^2_{\text{between}} = \max\left\{\frac{MS_B - MS_W}{\bar{n}}, 0\right\}, \quad \hat{\sigma}^2_{\text{within}} = MS_W \tag{A.7}$$

which in turn define the intraclass correlation coefficient (ICC) shown in Eq. (2).

Appendix B. Training protocol

The training protocol explained in Section 3.4 is presented in an algorithmic form in Algorithm 1.

Algorithm 1 Cross-validated training protocol for residual-boosted regression and group-aware anomaly detection.

Input: Grouped data $\{(x_i, y_i, g_i)\}_{i=1}^{N}$, anomaly threshold τ_{QI}, number of outer CV folds S, inner GBDT folds K

Output: Trained trunks and classifiers for each outer fold, Platt parameters, fusion coefficients, chosen threshold τ_{anom}, and test metrics per fold

for $s = 1 \ldots S$ **do** ▷ outer CV fold

Split groups into train/val/test for fold s (group-wise, no leakage).

Stage 1 - GBDT on raw features (cross-fitted on train)

Partition train indices into K folds $\{S_k\}_{k=1}^{K}$

for $k = 1 \ldots K$ **do**

Train GBDT on raw $\{(x_i, y_i) : i \notin S_k\}$

Obtain OOF predictions $\hat{y}_i^{\text{oof}} \leftarrow \text{GBDT}(x_i)$ for $i \in S_k$

Concatenate $\hat{y}^{\text{oof}}$ over all train rows

Stage 2 - Residuals (train)

$r_i \leftarrow y_i - \hat{y}_i^{\text{oof}}$ for all train rows

Stage 3 - Residual MLP on standardized features (train)

Fit standardization on train features; $\tilde{x}_i \leftarrow \text{Std}_{\text{train}}(x_i)$

Train MLP on $(\tilde{x}_i, r_i)$ with MSE; save trunk weights θ_{mlp}

Stage 4 - Initialize anomaly detector (train)

Initialize classifier trunk with θ_{mlp}

Stage 5 - Train anomaly detector (train)

Use group-aware sampler: many groups per batch, few rows per group

Train with focal BCE on $\tilde{x}_i$; use gradient clipping, OnecycleLR

Compute validation logits ℓ_j on *val* (std. with train stats)

Stage 6 - Platt calibration (val)

Fit (a, b) on *val* to minimize CE of $p_{AD,j} = \sigma(a\,\ell_j + b)$ vs labels

Stage 7 - Fusion model (val)

$\hat{y}_j \leftarrow \hat{y}_j^{\text{oof}} + \hat{r}_{\text{MLP}}(\tilde{x}_j)$

$m_j \leftarrow \tau_{QI} - \hat{y}_j, \quad s_j \leftarrow \text{logit}(p_{AD,j})$

Fit LR: $\text{logit}(p_{\text{FUSED},j}) = \beta_0 + \beta_1 s_j + \beta_2 m_j$ *(optional* $+\beta_3 s_j m_j$*)*

Stage 8 - Threshold selection (val)

Choose τ_{anom} on *val* to maximize F_1 (adaptable to application costs)

Stage 9 - Test evaluation (held-out test of fold s)

Standardize test; $p_{AD,t} = \sigma(a\,\ell_t + b)$

$m_t \leftarrow \tau_{QI} - \hat{y}_t$ with $\hat{y}_t = \hat{y}_t^{\text{oof}} + \hat{r}_{\text{MLP}}(\tilde{x}_t)$

Compute $p_{\text{FUSED},t}$ with β; binarize with τ_{anom}; record metrics

Appendix C. Implementation details, computational setup and hyperparameters

All experiments were implemented in Python using PyTorch for neural models and scikit-learn for gradient-boosted trees, Platt calibration, and logistic regression fusion. Data handling, preprocessing, and evaluation pipelines were built on NumPy and Pandas.

Training of the neural models was performed on an NVIDIA A100 GPU using the Multi-Instance GPU (MIG) feature. A single partition of 10 GB VRAM was allocated out of the seven available MIG instances on the card. This restricted memory footprint demonstrates that the proposed framework can be trained efficiently without requiring access to the full 80 GB of the device. Memory usage during training was dominated by batch construction, as the group-aware sampler maintains a balanced representation of multiple coils per batch..

Random seeds were fixed for data splits, network initialization, and training procedures, and deterministic operations were enabled where supported by PyTorch. Results are reproducible across runs, with only minor numerical differences possible due to non-deterministic GPU kernels in PyTorch/cuDNN.

Hyperparameters were fixed based on exploratory analysis and small independent grid searches rather than an extensive automated search. Learning rates, batch sizes, kernel sizes, and loss weights were held constant across CV folds to ensure comparability. Full details of the data

preprocessing settings, hyperparameter values and settings of models and training options, together with the software environment (Python version, PyTorch version, CUDA toolkit), are provided in tables Table C.1-C.9 to facilitate reproducibility.

Table C.1
Data preprocessing and grouping.

Item	Value	Notes
Grouping unit	group id	Coil id in the use case
Split policy	group-wise CV	No group leakage across folds
Standardization	StandardScaler	Fit on train only, applied to val and test
Windowing	lags = 3	Drops first lags rows per group
Window shape	$N_{features}, lags + 1)$	Axis-2 is $[t - l, \ldots, t]$
True anomaly label	$y \leq \tau_{QI}$	Process-defined lower bound

Table C.2
GBDT baseline (cross-fitted out-of-fold).

Hyperparameter	Value	Notes
Estimator	HistGradient BoostingRegressor	scikit-learn
Loss	squared_error	
Max iterations (trees)	100	
Learning rate	0.06	
Max leaf nodes	63	
L2 regularization	1×10^{-3}	
Early stopping	True	If used by sklearn HGB
Inner folds K	5	For cross-fitting OOF
Features	raw	No scaling for GBDT
Outputs	OOF predictions	Used to compute residuals

Table C.3
Residual MLP regressor.

Hyperparameter	Value	Notes
Input dim d	$N_{features}$	Number of engineered features
Hidden widths	(128, 64, 32, 16)	
Activation	ReLU	In all blocks
Dropout p	0.20	
Output	1	Scalar residual
Loss	MSE	
Optimizer	AdamW	
Learning rate - base	3×10^{-4}	Overridden by LR Scheduler
Weight decay	1×10^{-4}	
Epochs	400	
Warm-up fraction	0.05	Linear warm-up
LR scheduling	OneCycleLR	See Table C.4
Grad clip-norm	1.0	
Batch size	1024	MLP dataloader batch size
Standardization	yes	Fit on train only
Checkpoint	trunk weights	Used to init classifier trunk

Table C.4
OneCycleLR schedule - per step (train batches).

Hyperparameter	Value	Notes
Max learning rate	1×10^{-3}	Peak lr reached during the warm-up climb
Pct start	0.15	Fraction of steps for increasing lr - typical 0.3
Anneal strategy	cos	
Div factor	25	$lr_{init} = \max\_lr/div\_factor$
Final div factor	100	$lr_{final} = \max\_lr/final\_div\_factor$
Cycle momentum	True	If True - OneCycle also schedules optimizer momentum

Table C.5
Group-aware sampler and windowing.

Item	Value	Notes
Groups per batch	64	
Rows per coil	16	
Batch size	64×16	Constant per batch
Sampling	with replacement	Inside coil when short
Window lags	3	Drops first 3 rows per group

Table C.6
Group-aware anomaly detector with causal temporal encoder.

Component	Value	Notes
Temporal encoder	depth-wise Conv1d	in_ch = out_ch = d, groups = d
Kernel size	4	= lags +1 for lags = 3
Dilation, stride, padding	1, 1, 0	No padding for causality
Raw pathway	last time step	x_t per feature
Causal gate	learnable blend	LayerNorm on raw and conv; per-channel gate $a \in (0, 1)$
Gate init	0.10	Sigmoid of `logit_alpha` init
Trunk init	from MLP	Same widths: (128, 64, 32, 16), dropout 0.20
Group mean path	2-layer FC	$d \to \max(4, d/4) \to 1$ logit
Residual path	linear	$d \to 1$ logit on $z - \bar{z}_g$
Final logit	sum	Group logit + residual logit
Loss	focal BCE	$\gamma = 2.0$, $\alpha = 0.25$
Optimizer	AdamW	
Learning rate - base	3×10^{-4}	Overridden by LR Scheduler
Weight decay	1×10^{-4}	
Epochs	400	
LR scheduler	OneCycleLR	see Table C.4
Grad clip-norm	1.0	
Batching	group-aware	See Table C.5

Table C.7
Calibration and fusion.

Item	Value	Notes
Calibration	Platt scaling	Logistic regression on val
Platt params	(a, b)	Fitted by cross-entropy
Fusion inputs	$\text{logit}(p_{AD}),\ m = \tau_{QI} - \hat{y}$	No interaction by default
Fusion model	LogisticRegression	scikit-learn
Fusion solver	lbfgs	
Class weight	balanced	
Max iter	2000	
Threshold	τ_{anom}	Chosen on val to maximize F_1

Table C.8
Cross-validation and evaluation protocol.

Item	Value	Notes
Outer CV folds S	5	Group-wise
Inner folds K	5	For GBDT cross-fitting
Model selection	val loss	
Calibration set	validation	Platt and fusion fit on val
Test set	held-out outer fold	No refitting
Metrics - regression	RMSE	
Metrics - anomalies	Acc, Prec, Rec, F_1, F_β	β option per application

Table C.9
Compute and software environment.

Item	Value	Notes
GPU	NVIDIA A100 MIG	Single 10 GB partition
Frameworks	PyTorch, scikit-learn	Plus NumPy, pandas
CUDA toolkit	12.9	
PyTorch version	2.6.0	
Python version	3.9.16	
scikit-learn version	1.3.0	
Determinism	fixed seeds	Minor non-determinism possible on GPU kernels

References

[1] P.C. Deenen, J. Middelhuis, A. Akcay, I.J.B.F. Adan, Data-driven aggregate modeling of a semiconductor wafer fab to predict WIP levels and cycle time distributions, Flexible Serv. Manuf. J. 36 (2) (2024) 567–596. https://doi.org/10.1007/s10696-023-09501-1

[2] W.W.B. Goh, W. Wang, L. Wong, Why batch effects matter in omics data, and how to avoid them, Trends Biotechnol. 35 (6) (2017) 498–507. https://doi.org/10.1016/j.tibtech.2017.02.012

[3] Y. Yu, Y. Mai, Y. Zheng, L. Shi, Assessing and mitigating batch effects in large-scale omics studies, Genome Biol. 25 (254) (2024). https://doi.org/10.1186/s13059-024-03401-9

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, B.A. y Arcas, Communication-efficient learning of deep networks from decentralized data, in: Artificial Intelligence and Statistics, PMLR, 2017, pp. 1273–1282.

[5] T. Li, A.K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smith, Federated optimization in heterogeneous networks, Proc. Mach. Learn. Syst. 2 (2020) 429–450.

[6] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R.R. Salakhutdinov, A.J. Smola, Deep sets, Adv. Neural Inf. Process. Syst. 30 (2017), pp. 3394–3404.

[7] A. Tschalzev, P. Nitschke, L. Kirchdorfer, S. Lüdtke, C. Bartelt, H. Stuckenschmidt, Enabling mixed effects neural networks for diverse, clustered data using Monte Carlo methods, (2024). arXiv:2407.01115

[8] Y. Xiong, H.J. Kim, V. Singh, Mixed effects neural networks (menets) with applications to gaze estimation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7743–7752.

[9] G. Simchoni, S. Rosset, Integrating random effects in deep neural networks, (2023). arXiv:2206.03314

[10] M.-N. Tran, N. Nguyen, D. Nott, R. Kohn, Bayesian deep net GLM and GLMM, J. Comput. Graphical Stat. 29 (1) (2020) 97–113.

[11] W.E. Johnson, C. Li, A. Rabinovic, Adjusting batch effects in microarray expression data using empirical Bayes methods, Biostatistics 8 (1) (2007) 118–127.

[12] W. Jia, H. Li, R. Ali, K.P. Shanbhogue, W.R. Masch, A. Aslam, D.T. Harris, S.B. Reeder, J.R. Dillman, L. He, Investigation of ComBat harmonization on radiomic and deep features from multi-center abdominal MRI data, J. Imaging Inf. Med. 38 (2) (2025) 1016–1027.

[13] P. Murchan, P.Ó. Broin, A.-M. Baird, O. Sheils, S.P. Finn, Deep feature batch correction using ComBat for machine learning applications in computational pathology, J. Pathol. Inf. 15 (2024) 100396.

[14] F. Hu, A. Lucas, A.A. Chen, K. Coleman, H. Horng, R.W.S. Ng, N.J. Tustison, K.A. Davis, H. Shou, M. Li, et al., DeepComBat: a statistically motivated, hyperparameter-robust, deep learning approach to harmonization of neuroimaging data, Hum. Brain Mapp. 45 (11) (2024) e26708.

[15] T. Li, A.K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, V. Smithy, FedDANE: a federated newton-type method, in: 2019 53rd Asilomar Conference on Signals, Systems, and Computers, IEEE, 2019, pp. 1227–1231.

[16] F.T. Dong, F. Xue, L.X. Du, X.H. Liu, Effect of hot strip coiling temperature on microstructure and properties of boron-containing enamel steel, Mater. Res. Innovations 18 (Suppl. 4) (2014) S4–290–S4–294. https://doi.org/10.1179/1432891714Z.000000000694

[17] S. Oktay, P.E. Di Nunzio, M.C. Cesile, K. Davut, M.K. Şeşen, Effect of coiling temperature on the structure and properties of thermo-mechanically rolled S700MC steel, J. Min. Metall. Sect. B. 58 (3) (2022) 475–489.

[18] J. Xue, Z. Zhao, C. Bin, X. Liu, H. Wu, H. Li, W. Xiong, Effects of rolling and coiling temperature on the microstructure and mechanical properties of hot-rolled high strength complex phase steel, Mater. Res. Express 6 (9) (2019) 0965c8.

[19] M. Sun, Y. Xu, W. Du, Influence of coiling temperature on microstructure, precipitation behaviors and mechanical properties of a low carbon Ti micro-alloyed steel, Metals 10 (9) (2020) 1173.

[20] F. Peng, X. Gu, Y. Wang, Y. Xu, Y. Yu, Influence of coiling temperature on mechanical properties in hot rolling C-Mn-Si-Al steel, Procedia Manuf. 15 (2018) 52–58.

[21] Y. Yuasa, T. Yamane, M. Saito, M. Yoshino, Y. Miyai, R. Shimizu, New temperature control system for hot strip mill run out table, IFAC Proc. Volumes 23 (8) (1990) 143–148.

[22] H. Wu, J. Sun, W. Peng, D. Zhang, Analytical model for temperature prediction of hot-rolled strip based on symplectic space Hamiltonian system, Int. J. Heat Mass Transf. 213 (2023) 124350. https://doi.org/10.1016/j.ijheatmasstransfer.2023.124350

[23] Y. Zheng, N. Li, S. Li, Hot-rolled strip laminar cooling process plant-wide temperature monitoring and control, Contr. Eng. Pract. 21 (2013) 23–30. https://doi.org/10.1016/J.CONENGPRAC.2012.09.004

[24] W. Timm, K. Weinzierl, A. Leipertz, H. Zieger, G. Zouhar, Modelling of heat transfer in hot strip mill runout table cooling, Steel Res. 73 (2002) 97–104. https://doi.org/10.1002/SRIN.200200180

[25] H.B. Xie, Z.Y. Jiang, X.H. Liu, G.D. Wang, A.K. Tieu, Prediction of coiling temperature on run-out table of hot strip mill using data mining, J. Mater. Process. Technol. 177 (2006) 121–125. https://doi.org/10.1016/J.JMATPROTEC.2006.04.089

[26] S. Li, X. Li, Z. Deng, A new kind of model of laminar cooling: by LS-SVM and genetic algorithm, Commun. Comput. Inf. Sci. 472 (2014) 251–254. https://doi.org/10.1007/978-3-662-45049-9_41

[27] T.J. Sun, W.D. Yang, H. Gao, H.T. Mi, Coiling temperature prediction and application based on genetic-neural network on hot strip mill, Appl. Mech. Mater. 448–453 (2014) 3417–3420. https://doi.org/10.4028/WWW.SCIENTIFIC.NET/AMM.448-453.3417

[28] E.Y. Liu, W. Peng, N. Cao, S.R. Yu, J. Xu, L.G. Peng, D.H. Zhang, Prediction of coiling temperature of hot rolled strip based on BP neural network, Appl. Mech. Mater. 633–634 (2014) 679–683. https://doi.org/10.4028/WWW.SCIENTIFIC.NET/AMM.633-634.679

[29] B. Hu, Y. Zhang, S. Lu, F. Zhang, Q. Guo, J. Zhang, T. Yildirim, Temperature prediction for finish entry of hot strip mill based on a data-driven model, in: Proceedings of the 39th Chinese Control Conference (CCC), IEEE, 2020, pp. 2487–2493. https://doi.org/10.23919/CCC50068.2020.9188827

[30] D. Chen, R. Zhang, Z. Li, Y. Li, G. Yuan, Temperature distribution prediction in control-cooling process with recurrent neural network for variable-velocity hot-rolling strips, Int. J. Adv. Manuf. Technol. 120 (11–12) (2022) 7533–7546. https://doi.org/10.1007/s00170-022-09065-8

[31] H. Panjiar, M. Murugananth, Artificial neural network modeling of coiling temperature for dual-phase steel, in: International Conference on Mechanical Engineering, Springer, 2024, pp. 345–358.

[32] K. Zhang, Y. Wang, K. Peng, A hierarchical feature-fusion-based method for predicting the coiling temperature in a hot rolling mill process, Contr. Eng. Pract. 164 (2025) 106517. https://doi.org/10.1016/j.conengprac.2025.106517

[33] Z. Wang, J. Wang, S. Chen, Fault location of strip steel surface quality defects on hot-rolling production line based on information fusion of historical cases and process data, IEEE Access 8 (2020) 171240–171251.

[34] R.Y. Chen, W. Yuen, Oxide-scale structures formed on commercial hot-rolled steel strip and their formation mechanisms, Oxid. Met. 56 (1) (2001) 89–118.

[35] G.Y. Deng, H.T. Zhu, A.K. Tieu, L.H. Su, M. Reid, L. Zhang, P.T. Wei, Theoretical and experimental investigation of thermal and oxidation behaviours of a high speed steel work roll during hot rolling, Int. J. Mech. Sci. 131 (2017) 811–826.

[36] K. Min, K. Kim, S.K. Kim, D.-J. Lee, Effects of oxide layers on surface defects during hot rolling processes, Met. Mater. Int. 18 (2012) 341–348.

[37] A. Khakhar, J. Buckman, Neural regression for scale-varying targets, (2022). arXiv:2211.07447

[38] L. Nuyts, J. Davis, The when and how of target variable transformations, in: International Symposium on Intelligent Data Analysis, Springer, 2025, pp. 113–126.

[39] F.T. Lima, V.M.A. Souza, A large comparison of normalization methods on time series, Big Data Res. 34 (2023) 100407.

[40] K. Sechidis, G. Tsoumakas, I. Vlahavas, On the stratification of multi-label data, in: Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2011, Athens, Greece, September 5–9, 2011, Proceedings, Part III 22, Springer, 2011, pp. 145–158.

[41] D.C. Montgomery, Design and Analysis of Experiments, Wiley, 9th edition, 2017.

[42] S.R. Searle, G. Casella, C.E. McCulloch, Variance Components, Wiley, 1992.

[43] L. Grinsztajn, E. Oyallon, G. Varoquaux, Why do tree-based models still outperform deep learning on typical tabular data?, in: NeurIPS Datasets and Benchmarks Track, 2022. https://doi.org/10.48550/arXiv.2207.08815

[44] R. Shwartz-Ziv, A. Armon, Tabular data: deep learning is not all you need, Inf. Fusion 81 (2022) 84–90. https://doi.org/10.1016/j.inffus.2021.11.011

[45] G. Cybenko, Approximation by superpositions of a sigmoidal function, Math. Contr. Signals Syst. 2 (4) (1989) 303–314.

[46] K. Hornik, M. Stinchcombe, H. White, Multilayer feedforward networks are universal approximators, Neural Netw. 2 (5) (1989) 359–366.

[47] Y. Bengio, A. Courville, P. Vincent, Representation learning: a review and new perspectives, IEEE Trans. Pattern Anal. Mach. Intell. 35 (8) (2013) 1798–1828. https://doi.org/10.1109/TPAMI.2013.50

[48] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (6088) (1986) 533–536. https://doi.org/10.1038/323533a0

[49] J.H. Friedman, Greedy function approximation: a gradient boosting machine, Ann. Stat. (2001) 29 (05) 1189–1232.

[50] S. Alkhoury, E. Devijver, M. Clausel, M. Tami, É. Gaussier, G. Oppenheim, Smooth and consistent probabilistic regression trees, in: Advances in Neural Information Processing Systems 33 (NeurIPS 2020), 2020. https://proceedings.neurips.cc/paper/2020/file/8289889263db4a40463e3f358bb7c7a1-Paper.pdf.

[51] J. Platt, Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods, Technical Report MSR-TR-1999-28, Microsoft Research, 1999.

[52] M. Orabi, K.P. Tran, P. Egger, S. Thomassey, Anomaly detection in smart manufacturing: an adaptive adversarial transformer-based model, J. Manuf. Syst. 77 (2024) 591–611. https://doi.org/10.1016/j.jmsy.2024.09.021

[53] N. Shi, S. Guo, R. Al Kontar, Personalized feature extraction for manufacturing process signature characterization and anomaly detection, J. Manuf. Syst. 74 (2024) 435–448. https://doi.org/10.1016/j.jmsy.2024.04.002

[54] R. Zhao, R. Yan, Z. Chen, K. Mao, P. Wang, R.X. Gao, Deep learning and its applications to machine health monitoring, Mech. Syst. Signal Process. 115 (2019) 213–237. https://doi.org/10.1016/j.ymssp.2018.05.050