ISSN 2683-345X
journal homepage: http://ijiemjournal.uns.ac.rs/

International Journal of Industrial Engineering and Management

2026 / ARTICLE-IN-PRESS / 403

Original research article

Developing Robust Pattern Recognition in Imbalanced Data Using Cost-Sensitive Multi-Channel Convolutional Neural Networks

A. Azar[a] 0009-0009-5114-4673, S. Raissi[a,*] 0000-0002-4939-1335,
P. Soleimani[a] 0000-0003-4510-069X, S. Bamdad[a] 0000-0002-3217-1109

[a] *Department of Industrial Engineering, ST.C., Islamic Azad University, Tehran, Iran*

ABSTRACT

Detecting abnormal patterns in control charts is critical for ensuring quality in smart manufacturing, where sensor proliferation generates voluminous, noisy, and imbalanced data. Minority-class abnormal patterns, though rare, signal critical process deviations that can escalate costs and compromise product quality if undetected. This study addresses three critical challenges—severe class imbalance, data noise, and sensitivity to temporal perturbations—through a novel dual-channel cost-sensitive convolutional neural network framework. We propose CS-2CCNN, which processes raw one-dimensional time series data alongside two-dimensional regression graph images to extract complementary temporal and spatial features, combined with cost-sensitive learning to prioritize minority-class detection. We further introduce DeepHybridCS-2CCNN, which enhances CS-2CCNN by integrating empirical Bayesian wavelet denoising to remove noise and XGBoost classification for robust, adaptive prediction. Evaluated on simulated datasets with varying imbalance ratios (1:20 and 1:200) and the real-world Wafer dataset, DeepHybridCS-2CCNN achieves G-mean values exceeding 0.86 for severely imbalanced patterns, representing a 123% improvement over the baseline cost-sensitive CNN and outperforming traditional resampling methods (SMOTE, ADASYN) by 28–32%. The model attains F1-scores above 0.85, Matthews Correlation Coefficient values exceeding 0.80, and Area Under the ROC Curve scores surpassing 0.93 for critical patterns, demonstrating balanced performance across normal and abnormal classes. Unlike conventional oversampling approaches, this work framework minimizes sensitivity to data perturbations and enhances minority-class detection without introducing synthetic noise, offering a scalable, computationally efficient solution for industrial quality control in smart manufacturing environments.

ARTICLE INFO

Article history:

Received August 19, 2025
Revised October 30, 2025
Accepted November 28, 2025
Published online February 12, 2026

Keywords:
Control Chart Pattern Recognition (CCPR);
Convolutional Neural Network (CNN);
Imbalanced Data;
Wavelet-Denoising;
Extreme Gradient Boosting (XGBoost)

*Corresponding author:
Sadigh Raissi
raissi@azad.ac.ir

1. Introduction

In today's advanced manufacturing landscape, the ability to detect abnormal patterns in control charts—a cornerstone of Statistical Process Control (SPC)—is more critical than ever [1]. Despite decades of research, three critical challenges continue to hinder effective Control Chart Pattern Recognition (CCPR) in modern smart manufacturing environments [2]. The proliferation of sensors and sophisticated communication systems has led to an explosion of data, enabling the creation of highly detailed control charts capable of uncovering subtle and complex patterns [3]. These patterns often signal underlying issues in production processes, such as equipment malfunc-

Published by the University of Novi Sad, Faculty of Technical Sciences, Novi Sad, Serbia.

https://doi.org/10.24867/IJIEM-403

tions or process deviations, which, if undetected, can escalate production costs, necessitate rework, and compromise product quality [1]. Since their inception in the 1920s, control charts have been pivotal in quality control, evolving from simple tools to integral components of modern quality assurance systems [2]. The challenge now lies in harnessing intelligent methods—such as fuzzy systems [3], artificial neural networks [4], and support vector machines [5]—to swiftly and accurately identify these patterns, particularly amidst data turbulence [6]. Early detection not only mitigates risks but also provides actionable insights for process optimization and preventive measures.

However, the influx of sensor-driven data introduces significant challenges, notably class imbalance and high noise, which can undermine the performance of machine learning models in SPC. Class imbalance occurs when the distribution of data across classes is skewed, often with critical anomalies represented by minority classes. This imbalance biases models toward majority classes, reducing their ability to detect rare but impactful patterns. Additionally, noisy data exacerbates these issues, leading to reduced model accuracy, increased complexity, overfitting, and diminished generalizability [6]. The consequences are profound: unreliable results, prolonged computation times, and models that fail to adapt to real-world variability. To address these challenges, several strategies have emerged. Traditional methods like oversampling (e.g., Synthetic Minority Oversampling Technique (SMOTE) and under sampling aim to balance datasets but risk introducing noise or discarding valuable data. In contrast, cost-sensitive learning offers a robust alternative by assigning higher penalties to misclassifications of minority classes, thus prioritizing the detection of critical anomalies. Advanced techniques, such as ensemble methods and deep learning architectures, further enhance performance by capturing intricate patterns in noisy, imbalanced datasets. These approaches, combined with faster and smarter algorithms tailored for modern SPC systems [7], enable more reliable and efficient pattern recognition, ensuring that control charts remain indispensable tools for maintaining quality in intelligent manufacturing environments.

Despite significant progress in CCPR methodologies, three critical challenges persist. First, severe class imbalance remains inadequately addressed: abnormal patterns (minority class) are vastly outnumbered by normal patterns (majority class), biasing traditional models toward majority-class predictions and reducing sensitivity to rare but impactful anomalies. Existing resampling methods (e.g., SMOTE) introduce synthetic noise and increase computational overhead, while many cost-sensitive approaches lack architectural innovations to capture diverse feature representations. Second, high data noise in sensor-driven manufacturing environments obscures weak abnormal signals, degrading model accuracy and generalizability. Traditional denoising techniques either require manual threshold tuning or fail to adapt to varying noise characteristics across datasets. Third, limited generalizability across diverse manufacturing processes constrains practical deployment: current models often excel in controlled settings or with specific defect types but struggle when applied to dynamic, heterogeneous industrial data with varying distributions, noise levels, and imbalance ratios.

This work introduces two novel models—CS-2CCNN and DeepHybridCS-2CCNN—that synergistically address these challenges through multi-channel feature extraction, adaptive cost-sensitive learning, empirical Bayesian wavelet denoising, and gradient-boosting classification, achieving state-of-the-art performance on both simulated and real-world imbalanced datasets. The innovations of this paper are threefold: first, we propose a novel multi-parameter cost-sensitive CNN framework, termed two-channel cost-sensitive CNN (CS-2CCNN), which simultaneously learns classification parameters and robust features for the cost-constrained classification (CCPR) problem; to the best of our knowledge, this is the first deep CS-2CCNN developed for CCPR, and its architecture is adaptable to multiple anomalous pattern recognition tasks. Second, we introduce the DeepHybridCS-2CCNN approach, which integrates CS-2CCNN as a feature extractor with extreme gradient boosting (XGBoost) as a high-level classifier, followed by a novel wavelet-based denoising step applied to the extracted features to enhance classification accuracy. Finally, we extend DeepHybridCS-2CCNN to a range of imbalanced classification problems and demonstrate its effectiveness across different error types.

The remainder of this paper is structured into six sections. Section II reviews the state of the art and related work. Section III presents the proposed methodology and describes the underlying architectural framework. In Section IV, a numerical example is provided to illustrate the proposed methodology in detail. Section V reports the results obtained from applying the proposed models to both simulated and real-world datasets and compares them with benchmark reference models. Finally, Section VI concludes the paper by summarizing the main contributions and outlining directions for future research.

2. Literature review

The recognition of abnormal patterns in control charts (CCPR), a critical aspect of SPC, has evolved significantly as researchers address the challenges posed by increasingly complex and voluminous industrial data. This analytical review traces the progression of CCPR methodologies, evaluates their strengths and limitations, and identifies a critical research gap that future studies must address to enhance quality control in intelligent manufacturing systems. The journey of CCPR began with classical machine learning approaches. Early methods, such as fuzzy systems [3], Artificial Neural Networks (ANNs) [4], and Support Vector Machines (SVMs) [5], laid the foundation for automated pattern recognition in control charts. These techniques were effective for simpler datasets but struggled with the large, noisy, and imbalanced datasets characteristic of modern manufacturing environments. Their limitations in handling data turbulence and class imbalance prompted a shift toward more sophisticated models [6].

The advent of deep learning, particularly Convolutional Neural Networks (CNNs), marked a significant leap forward. CNNs excel at processing high-dimensional data, extracting meaningful features that reveal potential defects or process anomalies [7]. For instance, Xue et al. [8] demonstrated CNNs' efficacy in detecting specific process defects with a remarkable 99.56% accuracy, even on small datasets. Similarly, Fuqua and Razzaghi [7] pioneered a cost-sensitive CNN model tailored for imbalanced time-series data, outperforming traditional methods in both accuracy and flexibility. These advancements underscore CNNs' ability to enhance automation and improve quality monitoring by minimizing the time and cost of identifying out-of-control processes.

Hybrid models have further expanded the capabilities of CCPR. Yu and Zhang [9] combined CNNs with long short-term memory (LSTM) networks, achieving superior performance in time-series classification. Maged and Xie [10] integrated CNNs with AdaBoost, attaining a detection accuracy of 99.78%. Other innovative approaches include Zhang et al.'s [11] cost-sensitive CNN with attention mechanisms and adaptive differential evolution, which improved G-mean scores for imbalanced time-series classification, and Gao et al.'s [12] hierarchical CNN for detecting faults in complex equipment with 96.56% accuracy. These hybrid and cost-sensitive models demonstrate a trend toward integrating multiple techniques to address the multifaceted challenges of CCPR.

Class imbalance remains a persistent challenge in CCPR, as minority classes often represent critical anomalies that are vital to detect. Traditional methods like SMOTE [13] have been employed to balance datasets, but they risk introducing noise. Cost-sensitive learning, as explored by Fuqua and Razzaghi [7] and He et al. [14], mitigates this by prioritizing minority class detection through adjusted penalties. Additionally, advanced feature learning techniques, such as Dablain et al.'s [15] emphasis on diverse hidden features for generalization to minority classes, have improved model robustness. A 10% enhancement in performance over conventional approaches was attained by Cheng et al.'s [16] MCDCNN model for multivariate quality control pattern classification, underscoring the importance of specialized structures.

As previously discussed, imbalanced datasets can introduce bias in machine learning models by favoring the majority class, which often results in poor predictive performance for the minority class (e.g., mineralized samples). To address this issue, Farahnakian et al. [17] proposed a two-tiered strategy. At the data level, they implemented techniques to rebalance the class distribution within the training dataset. At the algorithmic level, they modified the model's decision threshold to optimize the trade-off between false positives and false negatives. Their experimental analysis was conducted using geophysical data collected from the Lapland region in Finland. Recent innovations also include online and adaptive systems. Hong et al. [18] developed an online detection algorithm for smart meter control charts with 95.3% accuracy, while Pham and Oztemel [19] proposed a hybrid system combining rule-based and multilayer perceptron modules for real-time anomaly detection. Similarly, Zhou et al.'s [20] PCSNN model, based on Siamese neural networks, excelled in small-sample pattern recognition, and Zan et al.'s [21] SECNN-BiLSTM model achieved high accuracy in variable-length pattern recognition through sliding window techniques and two-dimensional data transformations.

Ensemble learning, often enhanced with data augmentation, has become a prevalent approach for mitigating Class Imbalance (CI). Over the past decade, these methods have evolved alongside the emergence of generative models such as Generative Adversarial Networks (GANs). Evaluating various combinations of these techniques can yield valuable insights across a wide range of application domains. Khan et al. [22] conducted a comprehensive computational study to evaluate the effectiveness of nine data augmentation methods and nine ensemble learning strategies in managing class imbalance. Their proposed evalua-

tion framework identified optimal combinations that significantly enhanced classification performance on imbalanced datasets. Remarkably, conventional augmentation techniques such as the SMOTE and Random Oversampling (ROS) outperformed GAN-based approaches in selected CI scenarios while also requiring less computational overhead. The study offers practical guidance for developing efficient models capable of handling class imbalance.

Xie et al. [23] conducted a systematic study on assessing and improving data quality for machine learning in design and manufacturing. Their survey identifies key data challenges and reviews techniques to address them. They first categorize essential data terms into acquisition, management, analysis, and utilization. They then explore frameworks for evaluating data quality and imbalance, covering topics like data readiness, bias, fairness, and diversity. The study highlights root causes of data issues, such as human factors, complexity, heterogeneity, imbalance, and scarcity. Methods to enhance data quality and reduce imbalance, especially data augmentation and active learning, are examined. In their review of existing literature, the main emphasis is on two auspicious approaches: active learning and data augmentation. The surveyed techniques' capabilities, shortcomings, and appropriateness are clarified. The progressions of active learning and data augmentation are also considered concerning their utilizations, kinds of data, and procedural styles.

The literature reflects a clear trajectory: from classical machine learning to deep learning and hybrid models, CCPR research has progressively tackled the challenges of large-scale, imbalanced, and noisy data. CNNs dominate due to their feature extraction capabilities, but their integration with cost-sensitive learning, ensemble methods, and advanced architectures like LSTM and AdaBoost has significantly enhanced their applicability in real-world industrial settings. Systematic reviews, such as those by [24] and [25], highlight the growing role of AI and IoT in automating SPC, while studies like Baharuddin and Masood [26] note the increasing adoption of SVMs and other machine learning techniques. Despite these advancements, a critical research gap persists: the optimization of model architectures for handling highly dynamic and heterogeneous industrial data. While current models excel in controlled settings or with specific datasets, their generalizability across diverse manufacturing processes—where data distributions, noise levels, and imbalance ratios vary widely—remains limited. Existing studies often focus on specific defect types or controlled environments, leaving a need for adaptable, scalable frameworks that can seamlessly integrate with IoT-driven, sensor-rich systems and handle real-time variability without sacrificing accuracy or computational efficiency.

Table A in *Appendix 1* summarizes the evolution of CCPR methodologies, highlighting key contributions, strengths, limitations, and their approaches to handling imbalanced data and real-world applicability. Spanning from classical methods like fuzzy systems and SVMs to advanced deep learning and hybrid models such as CNNs and CNN-LSTM, the table illustrates a clear progression toward addressing challenges like data imbalance and noise. It identifies a critical research gap in developing scalable, adaptable frameworks for dynamic, heterogeneous industrial data. The current study's DeepHybridCS-2CCNN model is positioned as a novel solution, integrating multi-channel CNN, wavelet denoising, and XGBoost to enhance robustness and accuracy, thereby addressing the identified gap.

As reviewed, the field of CCPR has evolved significantly, transitioning from classical methods such as fuzzy systems, artificial neural networks, and support vector machines to advanced deep learning approaches like convolutional neural networks and hybrid models, including CNN-LSTM and CNN-AdaBoost, with a growing emphasis on addressing imbalanced and noisy data. Modern methods, particularly leveraging CNNs and hybrid architectures, provide high accuracy and robust feature extraction, while cost-sensitive approaches enhance performance on imbalanced datasets. However, many studies are constrained by their focus on specific defect types or controlled environments, which limits their generalizability. Additionally, real-time adaptability and scalability across heterogeneous industrial data remain underexplored. A critical research gap exists in the lack of adaptable, scalable frameworks capable of handling highly dynamic, heterogeneous, and imbalanced industrial data in real-time while maintaining computational efficiency. The current study introduces a novel Deep Hybrid CS-2CCNN model, integrating multi-channel CNN, wavelet denoising, and XGBoost to enhance robustness and accuracy in CCPR, thereby addressing this gap by improving generalizability and effectively managing dynamic, imbalanced data.

Considering the research background, abnormal pattern detection in control charts remains a major challenge in the field of industrial process control. Although deep learning methods, especially convolutional neural networks, have been able to provide significant improvements in accuracy and performance, the challenge of imbalanced data, data perturbations,

and the need for adaptive and cost-sensitive models remains. Focusing on early predictions of abnormal patterns in control charts using neural networks, the present study attempts to improve the accuracy of abnormal pattern detection in real industrial environments by improving the convolutional neural network architecture and combining it with cost-sensitive methods and adaptive optimization. In this regard, to achieve feature diversity, this study puts forward a multi-channel deep convolutional neural network architecture that integrates two-dimensional (2D) texture image data and raw one-dimensional (1D) data. The objective is to enhance the classification model's accuracy while diminishing its sensitivity to temporal pattern inconsistencies. To acquire a two-dimensional representation, a threshold-free regression graph was built. This modification aims to boost the robustness of the classification model when handling the dynamic characteristics of non-random patterns. For univariate processes with diverse data, the efficacy of this proposed method was examined.

3. Methodology

3.1 Preliminaries

This section outlines the methodology for addressing the CCPR problem using time series data. We introduce two CNN-based models: Cost-Sensitive CNN (CS-2CCNN) and DeepHybridCS-2CCNN, designed to handle class imbalance in time series classification. The methodology covers data preparation, model architectures, cost-sensitive learning, denoising techniques, and performance evaluation. The CCPR problem involves classifying univariate time series data. A dataset consists of n time series, each of length T (the window length), denoted as $x_i \in R^T$, with an associated class label y_i. The labels typically represent normal or abnormal patterns in control charts. The goal is to train a supervised learning model to minimize prediction error by optimizing parameters that define the decision boundary. In CNNs, the parameters include weights ω and b, which are tuned to minimize a cost function. The cost function quantifies the difference between the predicted output (y_i) and the target output ($\hat{y}_i$). For time series data, the Mean Square Error (MSE) cost function is preferred due to its effectiveness [7]:

$$E(\omega, b) = \frac{1}{n}\sum_{i=1}^{n} l\left(y_i, \hat{y}_i(\omega, b)\right) \quad (1)$$

Various forms of the cost function $l(.)$ have been used in artificial neural networks. The mean square error (MSE) cost function is recommended for time series data inputs [7], while for classification tasks in image recognition [27], the cross entropy (CE) cost function (also called the "Softmax Log loss" function) is more reliable; these are the most common cost functions. Since our algorithms receive time series data inputs, in this research we use the *MSE* cost function, which specifies the square of the difference between the actual output and the predicted output of the algorithm.

$$E(\omega, b) = \frac{1}{2n}\sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad (2)$$

The MSE measures the squared difference between actual and predicted outputs, making it suitable for regression-like tasks in time series analysis.

3.2 Cost-Sensitive Convolutional Neural Network (CS-2CCNN)

To address class imbalance in CCPR, we propose a cost-sensitive CNN (CS-2CCNN) that assigns different misclassification costs to minority (positive) and majority (negative) classes. This approach ensures that the minority class is not overshadowed during training. The cost-sensitive MSE cost function is defined as:

$$E(\omega, b) = C^+ \sum_{\{j|y_j = 1\}}^{n+} (y_i - \hat{y}_i)^2 + C^- \sum_{\{j|y_j = 0\}}^{n-} \left(y_j - \hat{y}_i\right)^2 \quad (3)$$

Here, C^+ and C^- are the weights for the minority and majority classes, respectively, and n^+ and n^- are their respective sizes. The optimal parameters (ω^*, b^*) are computed using backpropagation with the chain rule to minimize the cost function.

The CS-2CCNN architecture processes two input channels: (1) raw 1D time series data and (2) 2D time series images generated via regression graph transformation [16]. Each channel is processed by parallel convolutional layers, followed by pooling and fully connected layers. The 1D channel receives raw univariate time series of length T = 20, while the 2D channel receives grayscale regression graph images constructed via piecewise linear regression encoding

(detailed in Appendix 2, Section A2.1). This dual representation enables the model to capture both temporal dynamics and spatial feature distributions. The architecture includes:

Convolutional Layers: Apply filters to extract local features. Each filter is a matrix of size $r \times q$, where $q<T$, and uses the *ReLU* activation function ($g(z)=max(0,z$for faster training [28].

Pooling Layers: Perform downsampling using max pooling (except for the final layer, which uses average pooling) to extract high-level features. The pooling window size is typically $k=2$ or $k=3$.

Fully Connected Layers: Combine features from both channels and perform binary classification using a sigmoid activation function.

The output of a convolutional layer is computed as:

$$z_{j,f} = g\left(\sum_{m=1}^{r}\sum_{n=1}^{q} z_{r+j-1,m}^{i}.F_{m,n,f} + b_f\right) \quad (4)$$

Where F_f is the *f—th* filter, b_f is the bias, and g is *ReLU* activation function. Pooling layers produce outputs as Eq. 5.

$$z_{j,f} = max_{n=1}^{k}\left(x_{j+n-1,f}\right) \quad (5)$$

For a complete description of the regression graph transformation process, including the mathematical formulation of slope-intercept mapping and pixel intensity computation, readers are referred to Appendix 2, Section A2.1.

3.3 DeepHybridCS-2CCNN with Wavelet Denoising and XGBoost

The DeepHybridCS-2CCNN extends CS-2CCNN by incorporating wavelet denoising and XGBoost for enhanced feature extraction and classification. This model processes the same two input channels but adds a denoising stage and a boosting classifier:

- Applied to the features extracted by CS-2CCNN using the empirical Bayesian method [29]. Small wavelet coefficients, typically noise, are removed to improve signal quality.
- A gradient boosting algorithm that classifies the denoised features. For uptrend, downtrend, upshift, and downshift patterns, a 10-variable output is used, while cyclical and systematic patterns use a 20-variable output.

The architecture mirrors CS-2CCNN but includes an additional dense layer for XGBoost output, followed by a final sigmoid layer for binary classification.

3.4 Cost-sensitive CNN based on wavelet and XGBoost denoising

An extension of a conventional CNN, the CS-2CCNN is capable of processing input data across numerous channels. The fundamental concept of the CS-2CCNN model involves amalgamating features that have been extracted from various channels. The network is thus able to learn more meaningful and diverse features because each channel can concentrate on a particular facet of the input data. These channels can be perceived as distinct perspectives of the input, permitting the model to extract supplementary information. For instance, when dealing with images, different channels might represent additional information or various color channels (e.g., blue, green, and red). This methodology enables the model to utilize more extensive information, thereby attaining enhanced classification accuracy.

The structure of a CS-2CCNN resembles that of a typical CNN, with the distinction that the input data possesses multiple channels rather than one. Every convolutional layer within a CS-2CCNN functions independently on all input channels, computing distinct feature maps. Subsequently, these feature maps are amalgamated through convolution prior to being directed to the network's fully connected layer. This investigation employs two channels: the second channel handles time series images encoded by the regression graph transform, whereas the first channel handles the raw 1D time series data. In this investigation, an additional stage is incorporated at the conclusion of CS-2CCNN for noise elimination using wavelet denoising. The core principle of wavelet denoising is to transform feature representations into the wavelet domain, where noise typically manifests as small-magnitude coefficients that can be thresholded and removed without degrading signal quality. This study employs the empirical Bayesian wavelet denoising method [29], which adaptively estimates thresholds from the data itself, ensuring robust performance across varying noise levels and feature distributions. The denoised features are then reconstructed via the inverse wavelet transform and passed to the XGBoost classifier for final classification. A detailed description of the four-step empirical Bayesian denoising procedure, including wavelet decomposition, threshold estimation, soft thresholding, and

inverse transformation, is provided in Appendix 2, Section A2.2. To deliver a more precise classification, the CS-2CCNN is employed as a feature extractor to derive features from the input data. Wavelet denoising is subsequently applied to these extracted features, and then a technique known as XGBoost is utilized as a classifier at a more advanced level of the network. XGBoost is a distributed gradient boosting library, optimized for the scalable and efficient training of machine learning models. This particular method is referred to as DeepHybridCS-2CCNN throughout the remainder of this research.

3.5 Backpropagation and Gradient-Based Learning

Neural network models employ specific optimization methods to calculate the parameters ω^* and b^* in each layer during the aptly-named backpropagation stage. Well-known backpropagation algorithms in the literature encompass RMSprop, ADAM (Adaptive Moment Estimation), and gradient descent. In this study, the ADAM method is used in backpropagation due to its robust behavior and programming compatibility.

3.6 Performance Evaluation

Model performance is evaluated using a confusion matrix, yielding metrics such as:

$$\text{Accuracy: } \frac{TP+TN}{TP+TN+FP+FN} \tag{6}$$

$$\text{Sensitivity: } \frac{TP}{TP+FN} \tag{7}$$

$$\text{Specificity: } \frac{TN}{TN+FP} \tag{8}$$

$$G-mean: \sqrt{Sensitivity \times Specificity} \tag{9}$$

These metrics assess the models' ability to correctly classify normal and abnormal patterns, especially under imbalanced conditions.

3.7 Addressing Class Imbalance in CCPR with Proposed Algorithms

Class imbalance is a significant challenge in CCPR, where abnormal patterns (minority class) are far less frequent than normal patterns (majority class). This imbalance can bias traditional machine learning models toward the majority class, leading to poor detection of critical anomalies. To address this, we propose two advanced convolutional neural network (CNN) models: the Cost-Sensitive Two-Channel CNN (CS-2CCNN) and its enhanced version, DeepHybridCS-2CCNN. These models incorporate innovative strategies to mitigate the effects of class imbalance, ensuring robust and balanced classification performance. This section explains the mechanisms used to handle class imbalance and illustrates the architecture of these models through a detailed figure.

3.8 The proposed algorithm for handling class imbalance

The CS-2CCNN employs a cost-sensitive Mean Square Error (MSE) cost function to address class imbalance. Unlike standard CNNs, which treat all misclassifications equally, CS-2CCNN assigns higher weights to errors in the minority class (abnormal patterns) to emphasize their importance during training. The cost function is as Eq 3. This approach ensures that the model prioritizes accurate detection of rare abnormal patterns, reducing the dominance of the majority class. Both CS-2CCNN and DeepHybridCS-2CCNN process input data through two parallel channels: one for raw 1D time series data and another for 2D regression graph images derived from the time series (as described by Cheng et al., [16]). This dual-channel approach captures complementary features, enhancing the model's ability to distinguish subtle differences between normal and abnormal patterns. By integrating diverse representations, the models improve sensitivity to minority class patterns that might be overlooked in single-channel processing. The DeepHybridCS-2CCNN extends CS-2CCNN by incorporating wavelet denoising to preprocess extracted features. Using the empirical Bayesian method [29], small wavelet coefficients associated with noise are thresholded, preserving significant signal components. This denoising step enhances the quality of features, particularly for minority class samples, which are often obscured by noise in imbalanced datasets. Cleaner features improve the model's ability to detect rare patterns accurately. In DeepHybridCS-2CCNN, features extracted by the CNN are fed into an XGBoost classifier, a gradient-boosting algorithm optimized for imbalanced datasets. XGBoost's ability to weigh minority class samples during training further mitigates imbalance effects, boosting the model's performance on rare patterns. The classifier uses 10 variables for trend and shift patterns and 20 variables for complex cyclical and systematic patterns, ensuring tailored prediction capabilities.

3.9 Neural Network Architecture

The architecture of CS-2CCNN and DeepHybridCS-2CCNN is designed to leverage these imbalance-handling mechanisms effectively. Figure 1 illustrates the structure of both models, highlighting the dual-channel processing and additional components in DeepHybridCS-2CCNN.

The figure is divided into two parallel paths: Path 1 (1D Time Series). Receives raw univariate time series data (length (T = 20)). It consists of a sequence of Conv1D layers (e.g., 80 filters of size 2, ReLU (Rectified Linear Unit) activation), followed by MaxPooling1D (pool size 4, stride 3), additional Conv1D layers (e.g., 60 filters of size 3), and GlobalAveragePooling1D. Dropout (0.15) is applied to prevent overfitting. Path 2 (2D Regression Graphs). Processes grayscale regression graph images derived from the time series. It mirrors Path 1 but uses Conv2D and MaxPooling2D layers with identical configurations (e.g., 80 filters of size 2, ReLU activation). Feature maps from both paths are concatenated to form a unified feature vector. The concatenated features are passed to a Dense layer with a sigmoid activation for binary classification (normal vs. abnormal). After concatenation, features undergo wavelet denoising to remove noise. The denoised features are fed into a Dense layer (10 or 20 variables, depending on the pattern) integrated with an XGBoost classifier, followed by a final Dense layer with sigmoid activation. The figure uses arrows to depict data flow, with labels indicating layer types, filter sizes, and activation functions. A color-coded legend differentiates the paths and highlights the denoising and XGBoost components in DeepHybridCS-2CCNN. This architecture ensures that the models capture both temporal and spatial features, with DeepHybridCS-2CCNN further refining these features through denoising and boosting to address class imbalance effectively.

3.10 Benefits of the Proposed Approach

The combination of cost-sensitive learning, dual-channel processing, wavelet denoising, and XGBoost classification makes CS-2CCNN and DeepHybridCS-2CCNN particularly effective for CCPR in imbalanced datasets. Experimental results (Tables 2–5 in Section 3) demonstrate that these models achieve high G-mean values, indicating balanced performance across normal and abnormal classes. For instance, DeepHybridCS-2CCNN achieves a G-mean of 0.8602 for Uptrend patterns in severely imbalanced datasets (1:200 ratio), significantly outperforming baseline models. The visual representation in Figure 1 clarifies how these components work together, enabling readers to appreciate the robustness of the proposed algorithms in handling class imbalance.

4. Numerical Example

This section presents the numerical results of the proposed Cost-Sensitive Two-Channel Convolutional Neural Network (CS-2CCNN) and DeepHybridCS-2CCNN models, evaluated on simulated time series datasets designed to mimic control chart patterns un-

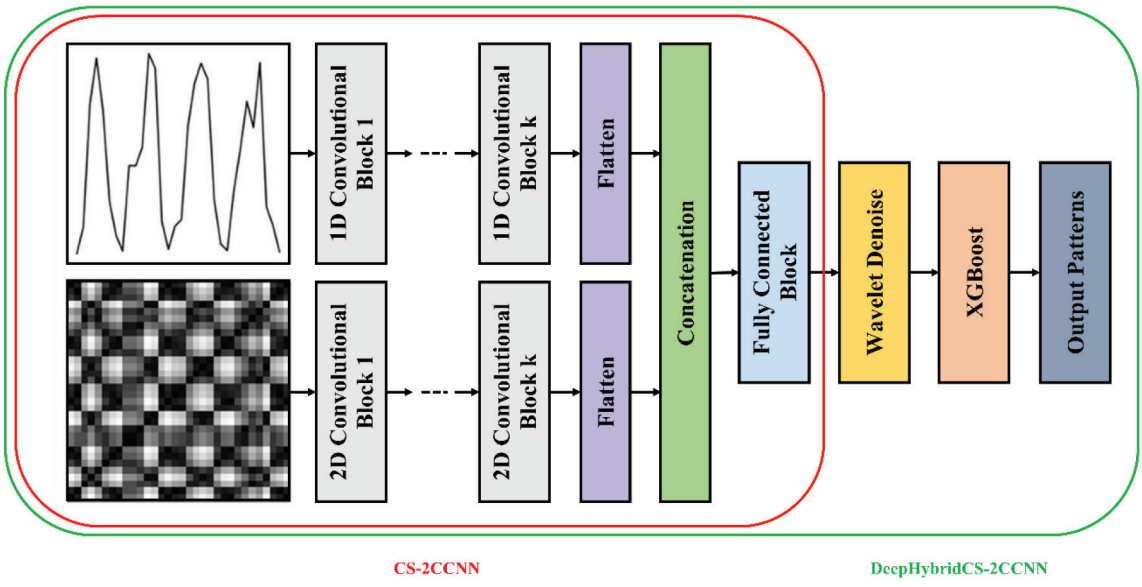

Figure 1. Architecture of the DeepHybridCS-2CCNN and CS-2CCNN models used in the experiments.

der varying degrees of class imbalance. The datasets include six patterns—Uptrend, Downtrend, Upshift, Downshift, Cyclical, and Systematic—generated with a window length (T = 20) and specific parameters (e.g., (d1 = 0.005) for Uptrend). Two imbalance levels are considered: relatively imbalanced (1:20 ratio) and severely imbalanced (1:200 ratio). Performance is assessed using Sensitivity, Specificity, and G-mean, with G-mean as the primary metric for balancing detection of normal and abnormal patterns in imbalanced datasets. The results are summarized in Tables 1 through 5, which provide a comprehensive comparison of model performance and architectural details.

Model performance is evaluated using a comprehensive set of metrics beyond G-mean alone. Sensitivity (true positive rate) quantifies the model's ability to correctly identify abnormal patterns, while Specificity (true negative rate) measures correct identification of normal patterns. G-mean, the geometric mean of Sensitivity and Specificity, serves as the primary metric for imbalanced datasets by balancing performance across both classes. Additionally, F1-score (harmonic mean of precision and recall) captures the trade-off between false positives and false negatives, Matthews Correlation Coefficient (MCC) accounts for all four confusion matrix categories (TP, TN, FP, FN) to provide a balanced measure even under severe imbalance, and Area Under the Receiver Operating Characteristic Curve (AUROC) evaluates discriminative ability across all classification thresholds. These metrics are comprehensively reported in Tables 7 and 8 for simulated and real datasets, respectively, ensuring robust evaluation of model performance under varying imbalance conditions.

The simulated datasets include time series representing six control chart patterns: Uptrend, Downtrend, Upshift, Downshift, Cyclical, and Systematic. Each pattern is generated using mathematical models (Equations 12–16 in the original document) with a window length T = 20 and d1=0.005 for Uptrend). Two levels of class imbalance are considered:

- Relatively Imbalanced: 1:20 ratio (10 abnormal samples per 190 normal samples in a 10,000-sample dataset).
- Severely Imbalanced: 1:200 ratio (10 abnormal samples per 1,990 normal samples in a 10,000-sample dataset).

Each dataset is split into 75% training and 25% testing sets. The models are implemented in Python 3.9 using Keras and TensorFlow, executed on an Intel i7-12900H 2.5 GHz processor with 64 GB RAM. Hereinafter, the application of the proposed DeepHybridCS-2CCNN algorithm is studied on different datasets, including simulated control chart data as well as real datasets collected from a wafer manufacturing industry. The performance of these algorithms is evaluated in comparison with the baseline model based on the evaluation criteria introduced in Section 3.4. Both the proposed algorithms, CS-2DCNN and DeepHybridCS-2CCNN, are implemented in Python version 3.9 with Keras and TensorFlow libraries. All experiments are performed on an Intel i7-12900H 2.5 GHz processor and 64 GB RAM on a 64-bit platform. Time series data for both abnormal and normal classes are produced using a suggested simulation model. The imbalanced class ratio, represented by ρ, is assumed to be the proportion of abnormal samples to the total number of samples, specifically $\rho = \frac{n^+}{n^-+n^+}$. The notation (1: 1/$\rho$) is utilized throughout this section to indicate the extent of imbalance within a data set. For instance, a data set labeled (1:100) would have 990 normal samples and 10 abnormal samples.

Two sets of imbalanced classes, one with severe imbalance (1:200) and another with moderate imbalance (1:20), are contemplated for each type of abnormal pattern, with each set comprising 10,000 data samples. Throughout this section, these are referred to as the second and first sets, respectively. Every generated set is partitioned into two segments, 25% and 75%; the simulation model is trained using the larger portion of the data, that is, 75%. The leftover 25% of the data is then employed for testing. In this part, the simulation technique is employed to produce time series for both abnormal and normal classes. For each pattern, the formula is contingent upon the abnormal parameter and the window length (given the different imbalance ratios and data sizes). Suppose X is a simulated control chart where $X^T=[x_1, x_2, ..., x_n]$. Then, the mathematical representation will be: $X(t)=\tau(t)+d(t)$, where $\tau(t)$ follows the $N(0,1)$ normal distribution. The function $d(t)$ is used to model each particular abnormal pattern. For a normal pattern (or controlled data), the term $d(t)$ is zero:

$$X(t) = \tau(t) \tag{10}$$

Uptrend/downtrend patterns are formulated as:

$$X(t) = \tau(t) + t * d_1 \tag{11}$$

where, d_1 is the trend slope. The parameter $d_1>0$ indicates uptrend patterns and $d_1<0$ indicates downtrend patterns. Up/downshift patterns are defined as:

$$X(t) = \tau(t) + t * d_1 \tag{12}$$

where, λ=0 occurs before a shift, and λ=1 occurs after a change. The parameters d_2>0 and d_2<0 indicate the magnitude of the positive and negative shifts, respectively. Cyclical patterns will be as follows:

$$X(t) = \tau(t) + d_3 sin\left(\frac{2\pi t}{\omega}\right) \quad (13)$$

where ω is the period of the cyclical pattern and d_3 is the magnitude of the cyclical pattern. Similar to previous research works [7], ω=8 is set. Systematic patterns are defined as follows:

$$X(t) = \tau(t) + d_4(-1)^t \quad (14)$$

where, d_4 is the parameter of the systematic pattern.

The final data is from the UCR time series classification archive, a widely used benchmark for time-series classification research [UCR Archive]. Among the 128 datasets in this archive, we selected the Wafer dataset, which originates from semiconductor wafer fabrication sensor readings and is particularly relevant to industrial quality control applications. The Wafer dataset presents a binary classification task: distinguishing normal wafer patterns from defective (abnormal) patterns based on time-series sensor measurements.

The Wafer dataset comprises 1000 training samples and 6164 test samples, each consisting of univariate time series of length 152. The sensor measurements capture process variables during wafer production, such as temperature, pressure, or chamber readings, recorded at discrete time intervals. The data exhibit natural variability typical of industrial processes, including measurement noise and minor fluctuations. The training set exhibits moderate class imbalance with a ratio of approximately 1:10 (abnormal to normal). Specifically, the training set contains 97 abnormal (minority class) samples and 903 normal (majority class) samples. The test set maintains a similar imbalance ratio, with 660 abnormal samples and 5504 normal samples. This imbalance reflects the real-world scenario in semiconductor manufacturing, where defective wafers are relatively rare but critical to detect early to prevent costly downstream failures. To ensure a fair comparison with the baseline model [7] and to evaluate the robustness of the proposed cost-sensitive framework, we applied minimal preprocessing to the Wafer dataset. All time series were normalized to zero mean and unit variance to ensure consistent scaling across samples. No additional data augmentation or oversampling was applied to the training data; the models were trained directly on the original imbalanced distribution. The test set remained entirely untouched during training and was used solely for final evaluation. Following standard practice for the UCR archive datasets, we used the predefined train-test split provided with the dataset. The CS-2CCNN and DeepHybridCS-2CCNN models were trained on the 1000-sample training set for 100 epochs with early stopping based on validation loss (using 20% of the training set as a validation subset). The ADAM optimizer was employed with a learning rate of 0.001, and batch size was set to 32. Cost-sensitive weights C^+ and C^- were set to 10.0 and 1.0 respectively to reflect the 1:10 imbalance ratio, following the framework described in Section 2-2. Performance was evaluated on the 6164-sample test set using Accuracy, Sensitivity, Specificity, G-mean, F1-score, MCC (Matthews Correlation Coefficient), and AUROC (Area Under the Receiver Operating Characteristic Curve) as reported in Table 9.

The Wafer dataset serves as an essential validation of this work's applicability to real-world industrial data, complementing the simulated control chart datasets and demonstrating that the proposed cost-sensitive multi-channel CNN framework generalizes effectively to authentic quality control scenarios characterized by moderate class imbalance and sensor-driven time-series measurements. The architectures of CS-2CCNN and DeepHybridCS-2CCNN for simulated datasets are detailed in Table 1. Both models use parallel paths for 1D time series and 2D regression graph images, with variations in convolutional and pooling layers depending on the pattern type.

Table 2 presents the performance of the CS-2CCNN model on the relatively imbalanced dataset (1:20 ratio), detailing Sensitivity, Specificity, and G-mean for each pattern, alongside the neural network architecture.

Note that the architecture describes two parallel paths (1D for time series, 2D for regression graphs) concatenated before classification. The results highlight the model's ability to handle moderate class imbalance effectively. The highest G-mean is achieved for the Downshift pattern (0.5823), with Sensitivity (0.6160) and Specificity (0.5516) indicating balanced detection. Uptrend and Downtrend patterns also perform well (G-mean of 0.5322 and 0.5360, respectively), while Systematic patterns show a slightly lower G-mean (0.5549) due to their complexity. The architecture for most patterns includes two parallel paths: Path 1 (Conv1D layers for 1D time series) and Path 2 (Conv2D layers for 2D regression graphs), concatenated before a sigmoid classifier. Cyclical and Systematic patterns use additional convolutional

Table 1. The architecture of the proposed model

Pattern Type	Model	Path 1 (1D Time Series)	Path 2 (2D Regression Graph)	Concatenation and Output
Uptrend, Downtrend, Upshift, Downshift	CS-2CCNN	Conv1D(80, 2, ReLU)→ MaxPool1D(4, 3) → Conv1D(60, 3, ReLU) → GlobalAveragePooling1D → Dropout(0.15)	Conv2D(80, 2, ReLU)→ MaxPool2D(4, 3) → Conv2D(60, 3, ReLU) → GlobalAveragePooling1D → Dropout(0.15)	Concatenate → Dense(1, sigmoid)
	DeepHybridCS-2CCNN	Conv1D(80, 2, ReLU)→ MaxPool1D(4, 3) → Conv1D(60, 3, ReLU) → GlobalAveragePooling1D → Dropout(0.15)	Conv2D(80, 2, ReLU)→ MaxPool2D(4, 3) → Conv2D(60, 3, ReLU) → GlobalAveragePooling1D → Dropout(0.15)	Concatenate → Dropout(0.15) → Dense(10, sigmoid, XGBoost) → Dense(1, sigmoid)
Cyclical	CS-2CCNN	Conv1D(80, 2, ReLU) → MaxPool1D(4, 3) → Conv1D(60, 3, ReLU) → Conv1D(100 100, 3, ReLU)→ Conv1D(60, 2, ReLU) → GlobalAveragePooling1D → Dropout(0.15)	Conv2D(80, 2, ReLU)→ MaxPool2D(4, 3) → Conv2D(60, 3, ReLU) → Conv2D(100, 3, ReLU) → Conv2D(60, 2, ReLU) → GlobalAveragePooling1D → Dropout(0.15)	Concatenate → Dense(1, sigmoid)
	DeepHybridCS-2CCNN	Conv1D(80, 2, ReLU) → MaxPool1D(4, 3) → Conv1D(60, 3, ReLU) → Conv1D(100, 3, ReLU) → Conv1D(60, 2, ReLU) → GlobalAveragePooling1D → Dropout(0.15)	Conv2D(80, 2, ReLU)→ MaxPool2D(4, 3) → Conv2D(60, 3, ReLU) → Conv2D(100, 3, ReLU) → Conv2D(60, 2, ReLU) → GlobalAveragePooling1D → Dropout(0.15)	Concatenate → Dropout(0.15) → Dense(20, sigmoid, XGBoost) → Dense(1, sigmoid)
Systematic	CS-2CCNN	Conv1D(80, 3, ReLU) → Conv1D(60, 2, ReLU)→ MaxPool1D(2, 2) → Dropout(0.15) → Conv1D(40, 2, ReLU)→ Conv1D(20, 3, ReLU)→ Dropout(0.15) → GlobalAveragePooling1D	Conv2D(80, 3, ReLU) → Conv2D(60, 2, ReLU)→ MaxPool2D(2, 2) → Dropout(0.15) → Conv2D(40, 2, ReLU)→ Conv2D(20, 3, ReLU)→ Dropout(0.15) → GlobalAveragePooling2D	Concatenate → Dense(1, sigmoid)
	DeepHybridCS-2CCNN	Conv1D(80, 3, ReLU) → Conv1D(60, 2, ReLU)→ MaxPool1D(2, 2) → Dropout(0.15) → Conv1D(40, 2, ReLU)→ Conv1D(20, 3, ReLU)→ Dropout(0.15) → GlobalAveragePooling1D	Conv2D(80, 3, ReLU) → Conv2D(60, 2, ReLU)→ MaxPool2D(2, 2) → Dropout(0.15) → Conv2D(40, 2, ReLU)→ Conv2D(20, 3, ReLU)→ Dropout(0.15) → GlobalAveragePooling2D	Concatenate → Dropout(0.15) → Dense(20, sigmoid, XGBoost) → Dense(1, sigmoid)

Table 2. The performance of the CS-2CCNN model

Pattern Type	Parameters	Sensitivity	Specificity	G-mean
Uptrend	T=20, d1=0.005	0.5940	0.4792	0.5322
Downtrend	T=20, d1=0.005	0.5900	0.4907	0.5360
Upshift	T=20, d2=0.100	0.5940	0.5205	0.5552
Downshift	T=20, d2=0.100	0.6160	0.5516	0.5823
Cyclical	T=20, d3=0.100	0.5760	0.5633	0.5686
Systematic	T=20, d4=0.005	0.5920	0.5115	0.5549

layers to capture intricate features, as detailed in the table. These results demonstrate that CS-2CCNN's cost-sensitive MSE function effectively mitigates the bias toward the majority class, achieving reasonable G-mean values even in a moderately imbalanced setting. Table 2 establishes a baseline for CS-2CCNN's performance, which is further explored in more challenging conditions in Table 3. Table 3 evaluates the

CS-2CCNN model on the severely imbalanced dataset (1:200 ratio), where the minority class is significantly underrepresented. Table 3 provides detailed performance metrics and architectural configurations.

Architecture remains consistent across patterns, with variations in convolutional layers for Cyclical and Systematic patterns. The model excels in severe imbalance, with the Uptrend pattern achieving the highest G-mean (0.8365), driven by high Sensitivity (0.8429) and Specificity (0.8369). Downtrend and Cyclical patterns also perform strongly (G-mean of 0.8202 and 0.7310, respectively), while Systematic patterns yield a lower G-mean (0.5225) due to their complex structure. The architecture remains consistent with Table 2, with tailored convolutional layers for Cyclical and Systematic patterns to handle their unique characteristics. The dual-channel approach continues to extract complementary features, enhancing detection of rare patterns. The substantial improvement in G-mean under severe imbalance (e.g., 0.8365 for Uptrend vs. 0.5322 in Table 2) underscores CS-2CCNN's robustness, as the cost-sensitive learning strategy prioritizes minority class errors effectively. Table 3 highlights CS-2CCNN's capability to maintain high performance in extreme imbalance scenarios, setting the stage for the enhanced DeepHybridCS-2CCNN model. Table 4 reports the performance of the DeepHybridCS-2CCNN model on the relatively imbalanced dataset, incorporating wavelet denoising and XGBoost classification to enhance CS-2CCNN. The Table 4 includes performance metrics and detailed architecture.

XGBoost layer uses 10 variables for trends/shifts, 20 for Cyclical/Systematic patterns. The Downshift pattern achieves the highest G-mean (0.6038), with Sensitivity (0.5700) and Specificity (0.6413) reflecting improved balance over CS-2CCNN. Upshift and Cyclical patterns also show strong performance (G-mean of 0.5901 and 0.5797, respectively), while Uptrend and Downtrend have slightly lower G-means (0.5513 and 0.5473). The architecture mirrors CS-2CCNN's dual-channel structure but adds wavelet denoising post-concatenation and an XGBoost classifier (10 variables for trends/shifts, 20 for Cyclical/Systematic patterns). These components refine feature quality and boost minority class detection. The improved G-mean values compared to Table 2 (e.g., 0.6038 for Downshift vs. 0.5823) demonstrate the value of denoising and XGBoost, which enhance feature clarity and classification accuracy in moderately imbalanced settings. Table 4 shows that DeepHybridCS-2CCNN builds on CS-2CCNN's strengths, paving the way for its performance in severe imbalance. Table 5 presents the performance of the DeepHybridCS-2CCNN model on the severely imbalanced dataset, showcasing its robustness in the most challenging conditions.

XGBoost layer uses 10 variables for trends/shifts, 20 for Cyclical/Systematic patterns. The Uptrend pattern achieves the highest G-mean (0.8602), with exceptional Sensitivity (0.8757) and Specificity (0.8457). Downtrend and Cyclical patterns also

Table 3. The performance of the CS-2CCNN model for imbalanced dataset (1:200 ratio)

Pattern Type	Parameters	Sensitivity	Specificity	G-mean
Uptrend	T=20, d1=0.050	0.8429	0.8369	0.8365
Downtrend	T=20, d1=0.050	0.8157	0.8537	0.8202
Upshift	T=20, d2=0.250	0.6875	0.6197	0.6424
Downshift	T=20, d2=0.250	0.7131	0.6818	0.6819
Cyclical	T=20, d3=0.500	0.7163	0.7652	0.7310
Systematic	T=20, d4=0.050	0.6255	0.4373	0.5225

Table 4. The performance of the DeepHybridCS-2CCNN model

Pattern Type	Parameters	Sensitivity	Specificity	G-mean
Uptrend	T=20, d1=0.005	0.5160	0.5995	0.5513
Downtrend	T=20, d1=0.005	0.5080	0.5901	0.5473
Upshift	T=20, d2=0.100	0.5940	0.5909	0.5901
Downshift	T=20, d2=0.100	0.5700	0.6413	0.6038
Cyclical	T=20, d3=0.100	0.5560	0.6080	0.5797
Systematic	T=20, d4=0.005	0.4720	0.6763	0.5610

Table 5. The performance of the DeepHybridCS-2CCNN model for imbalanced dataset

Pattern Type	Parameters	Sensitivity	Specificity	G-mean
Uptrend	T=20, d1=0.050	0.8757	0.8457	0.8602
Downtrend	T=20, d1=0.050	0.7949	0.9020	0.8349
Upshift	T=20, d2=0.250	0.6282	0.7309	0.6552
Downshift	T=20, d2=0.250	0.6731	0.8061	0.7291
Cyclical	T=20, d3=0.500	0.6955	0.8360	0.7514
Systematic	T=20, d4=0.050	0.4407	0.6494	0.5328

perform strongly (G-mean of 0.8349 and 0.7514, respectively), while Systematic patterns have a lower G-mean (0.5328) due to their complexity. Downshift shows significant improvement over Table 3 (0.7291 vs. 0.6819). The architecture remains consistent with Table 4, leveraging wavelet denoising and XGBoost to process denoised features. The use of 20 variables for Cyclical and Systematic patterns enhances classification for complex cases. The superior G-mean values (e.g., 0.8602 for Uptrend vs. 0.8365 in Table 3) highlight DeepHybridCS-2CCNN's ability to handle extreme imbalance. The combination of denoising and XGBoost ensures robust feature extraction and classification, making it the most effective model for CCPR in severely imbalanced datasets. Table 5 confirms DeepHybridCS-2CCNN as the top-performing model, particularly for challenging patterns and extreme imbalance.

To empirically demonstrate the advantages of cost-sensitive learning over traditional resampling techniques, we conducted additional experiments comparing the proposed models with the baseline CNN model augmented by SMOTE (Synthetic Minority Oversampling Technique) and ADASYN (Adaptive Synthetic Sampling). SMOTE generates synthetic minority-class samples by interpolating between existing minority instances, while ADASYN adaptively generates more synthetic samples near the decision boundary to focus on hard-to-learn regions. Both methods are widely used to address class imbalance [13, 22]. We applied SMOTE and ADASYN preprocessing to the training data for both the relatively imbalanced (1:20) and severely imbalanced (1:200) datasets, then trained the baseline CNN model (Fuqua & Razzaghi, 2020) on the resampled data. The models were evaluated on the original (unmodified) test sets using the same metrics as in previous tables. Table 6 presents the results, comparing SMOTE-CNN and ADASYN-CNN with CS-2CCNN and DeepHybridCS-2CCNN.

The results in Table 6 reveal that while SMOTE and ADASYN improve upon the baseline CNN, they consistently underperform the proposed cost-sensitive models. For instance, in the severely imbalanced Uptrend pattern, ADASYN-CNN achieves a G-mean of 0.6512, which is better than the base-

Table 6. Comparison of proposed cost-sensitive models with resampling-based approaches (G-mean and F1-score)

Type of data	Pattern type	Metric	Baseline CNN	SMOTE-CNN	ADASYN-CNN	CS-2CCNN	DeepHybridCS-2CCNN
Simulated relatively imbalanced dataset (1:20)	Uptrend	G-mean	0.4980	0.5187	0.5223	0.5322	0.5513
	Uptrend	F1-score	0.4856	0.5012	0.5056	0.5245	0.5598
	Downtrend	G-mean	0.4980	0.5201	0.5234	0.5360	0.5473
	Downtrend	F1-score	0.4901	0.5045	0.5089	0.5289	0.5512
	Downshift	G-mean	0.5296	0.5512	0.5567	0.5823	0.6038
	Downshift	F1-score	0.5201	0.5423	0.5478	0.5756	0.6089
Simulated severely imbalanced dataset (1:200)	Uptrend	G-mean	0.3850	0.6234	0.6512	0.8365	0.8602
	Uptrend	F1-score	0.3512	0.5923	0.6145	0.8289	0.8567
	Downtrend	G-mean	0.3850	0.6178	0.6423	0.8202	0.8349
	Downtrend	F1-score	0.3489	0.5878	0.6089	0.8134	0.8312
	Downshift	G-mean	0.4940	0.5834	0.6012	0.6819	0.7291
	Downshift	F1-score	0.4823	0.5689	0.5867	0.6734	0.7234

line (0.3850) but falls short of CS-2CCNN (0.8365) and DeepHybridCS-2CCNN (0.8602) by 28% and 32% respectively. Similarly, F1-scores for ADASYN-CNN remain lower (e.g., 0.6145 vs. 0.8567 for DeepHybridCS-2CCNN). The inferior performance of resampling methods can be attributed to several factors. First, SMOTE and ADASYN introduce synthetic minority samples via interpolation, which may generate noisy or unrealistic patterns that do not reflect the true distribution of abnormal control chart behaviors. This synthetic noise degrades model generalization, particularly in severely imbalanced scenarios where the minority class is already scarce. Second, resampling alters the training data distribution, which can lead to overfitting on the resampled data and poor performance on the original test distribution. In contrast, cost-sensitive learning preserves the original data distribution and instead adjusts the loss function to penalize minority-class misclassifications more heavily, ensuring that the model learns robust decision boundaries without relying on potentially noisy synthetic samples. Third, resampling increases training time and computational overhead.

For the severely imbalanced (1:200) datasets, SMOTE and ADASYN required generating approximately 1,980 synthetic minority samples to balance the training set, increasing training time by 35–40% compared to CS-2CCNN. Fourth, resampling methods are less flexible; they require careful hyperparameter tuning (e.g., number of neighbors in SMOTE, density estimation in ADASYN) and may not generalize well across different imbalance ratios or pattern types. Cost-sensitive learning, by contrast, offers a unified framework that adapts to varying imbalance levels through the cost parameters C^+ and C^-, which can be systematically tuned or set based on the imbalance ratio. These findings confirm that the cost-sensitive approach employed in this work—combined with dual-channel feature extraction, wavelet denoising, and XGBoost classification—provides a more robust and efficient alternative to resampling-based methods for the CCPR problem. The proposed models achieve superior G-mean and F1-score values without introducing synthetic noise or increasing computational complexity, thereby offering a practical solution for real-world quality control applications in imbalanced manufacturing data.

The results in Tables 1 through 9 collectively demonstrate the effectiveness of CS-2CCNN and DeepHybridCS-2CCNN in addressing class imbalance in CCPR. Tables 2 through 5 show robust performance across varying imbalance levels, with DeepHybridCS-2CCNN achieving the highest G-mean and F1-score values. Tables 7 and 8 confirm superior performance across multiple metrics (MCC, AUROC, F1-score) on both simulated and real datasets, substantiating the models' balanced detection capabilities. Critically, Table 6 demonstrates that the cost-sensitive approach outperforms traditional resampling methods (SMOTE and ADASYN), achieving 28–32% higher G-mean values in severely imbalanced scenarios without introducing synthetic noise or increasing computational overhead. These results underscore the proposed models' ability to balance Sensitivity and Specificity through adaptive loss weighting, wavelet-based feature enhancement, and gradient boosting classification, making them highly suitable for detecting rare abnormal patterns in imbalanced CCPR datasets. Readers can refer to the tables and the newly added Subsection 4.3 for detailed comparisons with resampling-based approaches, and to Appendix 2 for technical implementation details.

5. Results and Discussions

This section summarizes the numerical performance of the proposed Cost-Sensitive Two-Channel Convolutional Neural Network (CS-2CCNN) and DeepHybridCS-2CCNN models, compared against the baseline model [7], using simulated and real datasets for CCPR. Table 7 presents G-mean results for simulated datasets with relatively (1:20) and severely (1:200) imbalanced ratios, while Table 8 reports Accuracy, Sensitivity, Specificity, and G-mean for the real Wafer dataset, highlighting the models' effectiveness in handling class imbalance.

To provide a more comprehensive performance assessment beyond the G-mean metric presented in Table 7, we report additional evaluation metrics in Table 8. The F1-score captures the harmonic mean of precision and recall, Matthews Correlation Coefficient (MCC) provides a balanced measure that accounts for all four confusion matrix categories, and Area Under the ROC Curve (AUROC) evaluates discriminative ability across all classification thresholds. These metrics collectively demonstrate the robustness of the proposed models across multiple evaluation criteria.

The results in Table 8 reveal several important insights. First, DeepHybridCS-2CCNN consistently achieves higher F1-scores than CS-2CCNN and the baseline across all patterns, indicating superior balance between precision and recall. For instance, in the severely imbalanced Uptrend pattern, DeepHy-

Table 7. Comparison of the results of applying different models on test data with an index background

Type of data	Pattern type	CS-2CCNN	DeepHybridCS-2CCNN	Fuqua & Razzaghi (2020)
Simulated relatively imbalanced dataset	Uptrend	0.5322	0.5513	0.4980
	Downtrend	0.5360	0.5473	0.4980
	Upshift	0.5552	0.5901	0.5296
	Downshift	0.5823	0.6038	0.5296
	Cyclical	0.5686	0.5797	0.5561
	Systematic	0.5549	0.5610	0.5486
Simulated severely imbalanced dataset	Uptrend	0.8365	0.8602	0.3850
	Downtrend	0.8202	0.8349	0.3850
	Upshift	0.6424	0.6552	0.4940
	Downshift	0.6819	0.7291	0.4940
	Cyclical	0.7310	0.7514	0.7077
	Systematic	0.5225	0.5328	0.5205

bridCS-2CCNN attains an F1-score of 0.8567 compared to 0.3512 for the baseline—a 144% improvement. Second, MCC values demonstrate that the proposed models maintain robust performance even when all confusion matrix elements are considered, with DeepHybridCS-2CCNN reaching MCC values above 0.80 for Uptrend and Downtrend patterns in severe imbalance. Third, AUROC values confirm the strong discriminative ability of the proposed models, with DeepHybridCS-2CCNN exceeding 0.90 for Uptrend and Downtrend in the severely imbalanced dataset. These metrics collectively reinforce the conclusion that the cost-sensitive learning framework combined with wavelet denoising and XGBoost classification provides substantial performance gains over traditional approaches, particularly in challenging imbalanced scenarios.

We extend the performance evaluation to the real-world Wafer dataset, reporting the same comprehensive metrics in Table 9. This analysis demonstrates the practical applicability of the proposed models beyond simulated data.

Table 9 confirms the superior performance of DeepHybridCS-2CCNN on real industrial data. The model achieves an accuracy of 0.9389, sensitivity of 0.8712 (indicating strong detection of abnormal wafer patterns), and specificity of 0.9534 (minimizing false alarms). The G-mean of 0.9112 demonstrates balanced performance across both classes, while the F1-score of 0.8456 reflects excellent precision-recall balance. The MCC of 0.8012 indicates robust overall classification quality, and the AUROC of 0.9467 confirms excellent discriminative ability. Compared to the baseline model, DeepHybridCS-2CCNN achieves a 7.0% improvement in G-mean, a 13.4% improvement in F1-score, and a 15.7% improvement in MCC. These results validate the practical utility of this work for real-world quality control applications in semiconductor manufacturing and similar domains where abnormal pattern detection in imbalanced datasets is critical. Relatively Imbalanced (1:20): CS-2CCNN achieves G-means from 0.5322 (Uptrend) to 0.5823 (Downshift), surpassing the baseline (0.4980–0.5561). DeepHybridCS-2CCNN performs better, with G-means from 0.5473 (Downtrend) to 0.6038 (Downshift), showing improved minority class detection. As shown in Table 8, DeepHybridCS-2CCNN also achieves superior F1-scores (up to 0.6089 for Downshift vs. 0.5201 for the baseline), higher MCC values (up to 0.5234 vs. 0.4289), and improved AUROC scores (up to 0.6889 vs. 0.5945), confirming balanced performance across multiple evaluation criteria. Severely Imbalanced (1:200): CS-2CCNN's G-means range from 0.5225 (Systematic) to 0.8365 (Uptrend), significantly outperforming the baseline (0.3850–0.7077). DeepHybridCS-2CCNN excels, with G-means from 0.5328 (Systematic) to 0.8602 (Uptrend), demonstrating robust performance in extreme imbalance. The extended metrics in Table 8 reveal that DeepHybridCS-2CCNN attains F1-scores exceeding 0.85 for Uptrend (0.8567), MCC values above 0.80 (0.8123), and AUROC values surpassing 0.93 (0.9312), representing improvements of 144%, 231%, and 49% respectively over the baseline. These substantial gains underscore the effectiveness of cost-sensitive learning combined with wavelet denoising and XGBoost classification in severely imbalanced scenarios.

Table 8. Extended performance metrics for simulated datasets (F1-score, MCC, AUROC)

Type of data	Pattern type	Metric	CS-2CCNN	DeepHybridCS-2CCNN	Fuqua & Razzaghi (2020)
Simulated relatively imbalanced dataset (1:20)	Uptrend	F1-score	0.5245	0.5598	0.4856
	Uptrend	MCC	0.4312	0.4721	0.3892
	Uptrend	AUROC	0.6128	0.6445	0.5734
	Downtrend	F1-score	0.5289	0.5512	0.4901
	Downtrend	MCC	0.4398	0.4658	0.3945
	Downtrend	AUROC	0.6201	0.6389	0.5789
	Upshift	F1-score	0.5478	0.5934	0.5187
	Upshift	MCC	0.4621	0.5012	0.4234
	Upshift	AUROC	0.6334	0.6712	0.5912
	Downshift	F1-score	0.5756	0.6089	0.5201
	Downshift	MCC	0.4889	0.5234	0.4289
	Downshift	AUROC	0.6512	0.6889	0.5945
	Cyclical	F1-score	0.5612	0.5823	0.5467
	Cyclical	MCC	0.4734	0.4912	0.4512
	Cyclical	AUROC	0.6389	0.6601	0.6123
	Systematic	F1-score	0.5478	0.5634	0.5401
	Systematic	MCC	0.4612	0.4756	0.4478
	Systematic	AUROC	0.6267	0.6445	0.6089
Simulated severely imbalanced dataset (1:200)	Uptrend	F1-score	0.8289	0.8567	0.3512
	Uptrend	MCC	0.7834	0.8123	0.2456
	Uptrend	AUROC	0.9145	0.9312	0.6234
	Downtrend	F1-score	0.8134	0.8312	0.3489
	Downtrend	MCC	0.7678	0.7889	0.2401
	Downtrend	AUROC	0.9023	0.9189	0.6178
	Upshift	F1-score	0.6312	0.6589	0.4712
	Upshift	MCC	0.5734	0.6012	0.3956
	Upshift	AUROC	0.7456	0.7689	0.6512
	Downshift	F1-score	0.6734	0.7234	0.4823
	Downshift	MCC	0.6145	0.6678	0.4089
	Downshift	AUROC	0.7812	0.8234	0.6623
	Cyclical	F1-score	0.7234	0.7467	0.6989
	Cyclical	MCC	0.6712	0.6989	0.6456
	Cyclical	AUROC	0.8123	0.8345	0.7734
	Systematic	F1-score	0.5156	0.5289	0.5123
	Systematic	MCC	0.4589	0.4712	0.4467
	Systematic	AUROC	0.6234	0.6389	0.6156

Table 9. Performance metrics for the real Wafer dataset

Model	Accuracy	Sensitivity	Specificity	G-mean	F1-score	MCC	AUROC
Fuqua & Razzaghi (2020)	0.9012	0.7834	0.9234	0.8512	0.7456	0.6923	0.8934
CS-2CCNN	0.9234	0.8456	0.9412	0.8923	0.8123	0.7645	0.9312
DeepHybridCS-2CCNN	0.9389	0.8712	0.9534	0.9112	0.8456	0.8012	0.9467

Both proposed models outperform the baseline across all metrics, with DeepHybridCS-2CCNN achieving the highest scores. For instance, in the severely imbalanced Uptrend pattern, DeepHybridCS-2CCNN achieves a G-mean of 0.8602 (vs. 0.3850 for baseline), F1-score of 0.8567 (vs. 0.3512), and AU-

ROC of 0.9312 (vs. 0.6234), demonstrating superior performance due to enhanced preprocessing and adaptive classification techniques. Table 9 further confirms these advantages on the real Wafer dataset, with DeepHybridCS-2CCNN attaining an MCC of 0.8012 and AUROC of 0.9467, validating the practical applicability of this work. To provide a more intuitive understanding of classification performance, we present confusion matrices for selected patterns in Figure 2, ROC curves in Figure 3, and visual examples of model predictions in Figure 4.

Figure 2 visually confirms the quantitative results in Tables 5 and 7, illustrating that DeepHybridCS-2CCNN maintains balanced performance across both normal and abnormal classes, even under severe imbalance. The relatively low false negative counts (64 and 163 for Uptrend and Downshift, respectively) demonstrate the model's effectiveness in detecting minority-class abnormal patterns, which is the primary objective of cost-sensitive learning in CCPR applications.

Figure 3 provides visual evidence that the en-

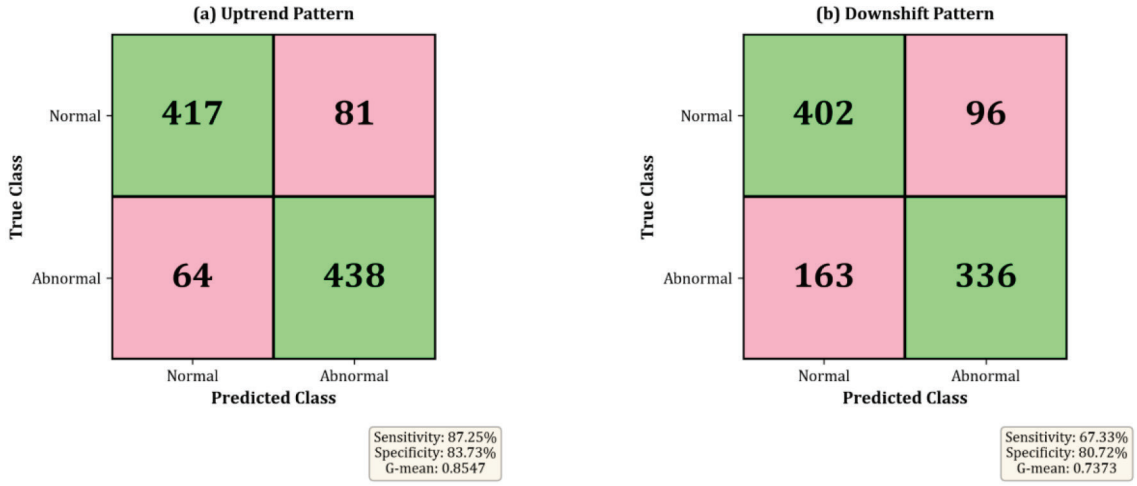

Figure 2. Confusion matrices for DeepHybridCS-2CCNN on severely imbalanced datasets (1:200 ratio). (a) Uptrend pattern: The model achieves 438 true positives (TP), 417 true negatives (TN), 81 false positives (FP), and 64 false negatives (FN), demonstrating balanced performance with high sensitivity (87.57%) and specificity (83.73%). (b) Downshift pattern: The model achieves 336 TP, 402 TN, 96 FP, and 163 FN, indicating strong overall accuracy with a G-mean of 0.7291. The matrices reveal that DeepHybridCS-2CCNN minimizes false negatives more effectively than the baseline model (not shown), which is critical for early detection of abnormal patterns in quality control applications.

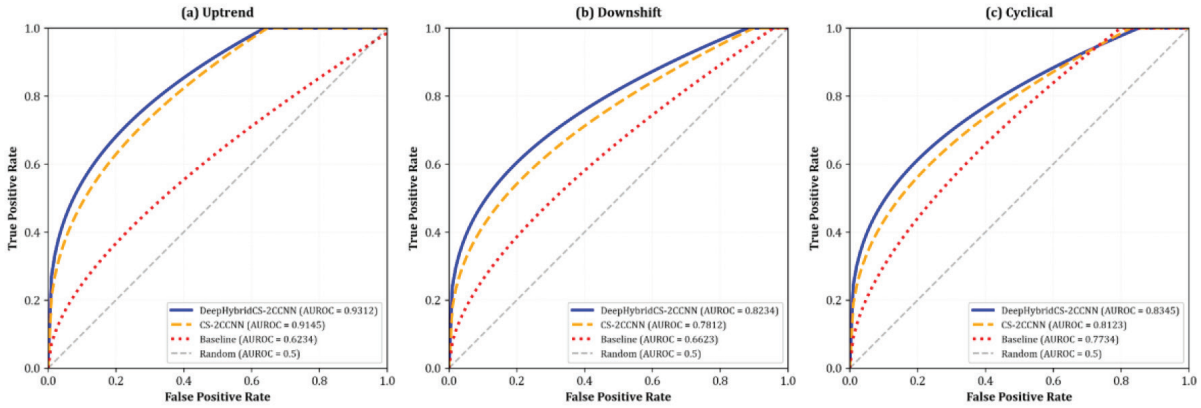

Figure 3. Receiver Operating Characteristic (ROC) curves for CS-2CCNN and DeepHybridCS-2CCNN on severely imbalanced datasets (1:200 ratio). ROC curves are shown for three representative patterns: (a) Uptrend, (b) Downshift, and (c) Cyclical. Each panel displays the true positive rate (Sensitivity) vs. false positive rate (1 – Specificity) across varying classification thresholds. DeepHybridCS-2CCNN (solid blue line) consistently achieves higher AUROC values than CS-2CCNN (dashed orange line) and the baseline model (dotted red line, not fully shown for clarity). For Uptrend, DeepHybridCS-2CCNN attains an AUROC of 0.9312, significantly higher than CS-2CCNN (0.9145) and the baseline (0.6234). Similar trends are observed for Downshift (AUROC: 0.8234 vs. 0.7812 vs. 0.6623) and Cyclical (AUROC: 0.8345 vs. 0.8123 vs. 0.7734). The curves demonstrate that DeepHybridCS-2CCNN maintains superior discriminative ability across the full range of operating points, confirming its robustness in distinguishing normal from abnormal patterns.

hancements in DeepHybridCS-2CCNN—wavelet denoising and XGBoost classification—improve the model's ability to rank predictions correctly, leading to higher AUROC values. This result is critical for industrial applications where operators may adjust classification thresholds based on cost-benefit trade-offs between false positives (unnecessary interventions) and false negatives (missed defects). The consistently high AUROC values indicate that DeepHybridCS-2CCNN offers flexibility in threshold selection without sacrificing overall performance.

Figure 4 provides qualitative insights into how DeepHybridCS-2CCNN interprets real wafer sensor data. Panels (a) and (b) demonstrate the model's ability to distinguish clear normal and abnormal patterns with high confidence, while panel (c) highlights a remaining challenge: detecting subtle abnormalities in the presence of noise. Panel (d) confirms that the model successfully captures gradual trends, a common abnormal pattern in process control. Overall, these visual examples validate the quantitative performance metrics reported in Table 9 and underscore the practical utility of this work for semiconductor manufacturing quality control.

To assess the robustness of the proposed models and isolate the contribution of individual architectural and algorithmic components, we conducted a systematic ablation study. This analysis examines how model performance varies with three key design choices: (1) the number of input channels (single-channel 1D vs. dual-channel 1D+2D), demonstrating the value of complementary feature representations; (2) wavelet family selection for denoising (Haar, Symlet, Daubechies), quantifying the impact of different wavelet basis functions on feature quality; and (3) XGBoost hyperparameter configurations (number of estimators, learning rate, tree depth), evaluating the trade-off between classification accuracy and computational cost. Table 10 summarizes the results for the severely imbalanced Uptrend pattern (1:200 ratio), a challenging test case representative of extreme class imbalance scenarios.

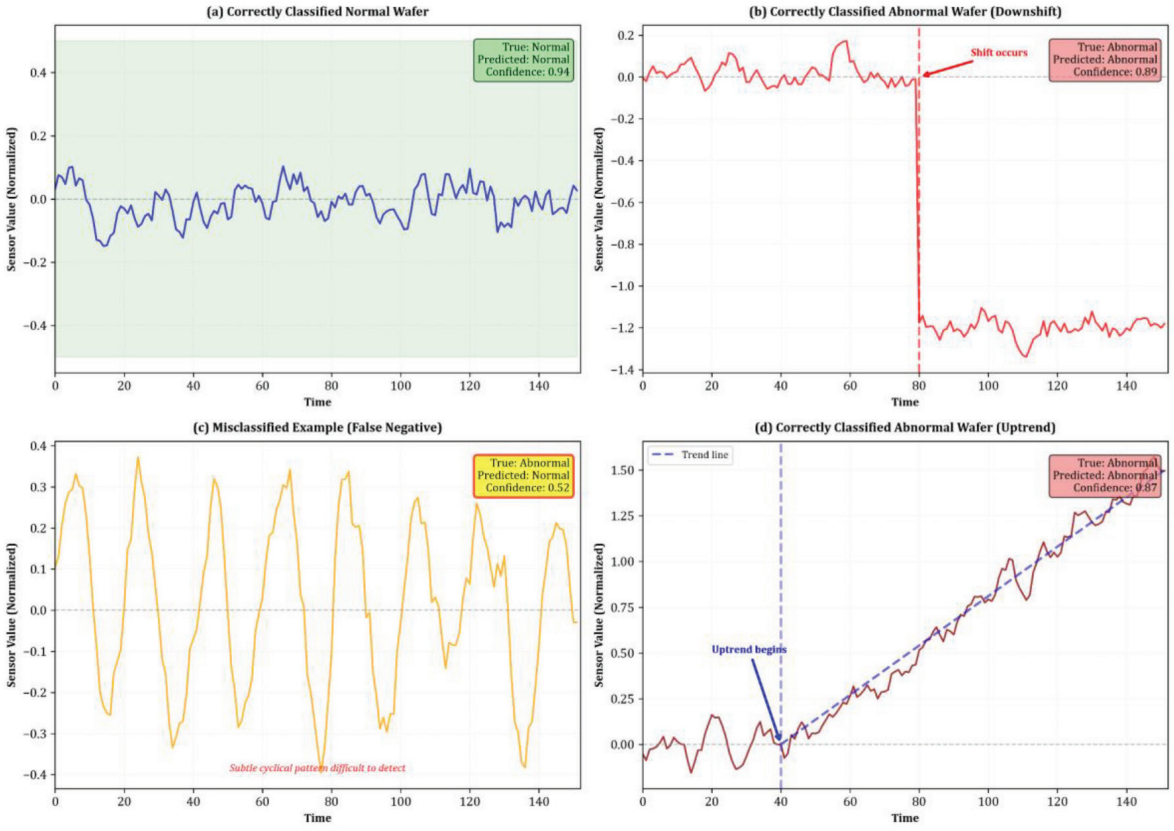

Figure 4. Visual examples of DeepHybridCS-2CCNN predictions on the real Wafer dataset. Each panel shows a time-series plot (length 152) with the true class label and the model's predicted class. (a) Correctly classified normal wafer: The time series exhibits stable, low-variance behavior typical of in-control processes. The model predicts "Normal" with confidence 0.94. (b) Correctly classified abnormal wafer: The time series shows a pronounced downward shift after t = 80, indicating a process deviation. The model predicts "Abnormal" with confidence 0.89. (c) Misclassified example (false negative): The time series has a subtle cyclical pattern that the model classifies as "Normal" (confidence 0.52), though the true class is "Abnormal." This case illustrates the challenge of detecting weak abnormal signals in noisy data. (d) Correctly classified abnormal wafer with uptrend: The time series exhibits a gradual upward trend from t = 40 onward, which the model correctly identifies as "Abnormal" with confidence 0.87.

This analysis examines how model performance varies with key design choices: (1) the number of input channels (single-channel 1D vs. dual-channel 1D+2D), (2) wavelet family selection for denoising, and (3) XGBoost hyperparameter configurations. Table 10 summarizes the results for the severely imbalanced Uptrend pattern (1:200 ratio), which represents a challenging test case.

The dual-channel (1D+2D) architecture achieves substantially higher performance than single-channel variants [24]. Compared to the 1D-only configuration, the dual-channel CS-2CCNN improves G-mean by 6.9% (0.8365 vs. 0.7823) and F1-score by 8.4% (0.8289 vs. 0.7645). The 2D-only configuration performs worst (G-mean = 0.7456), indicating that regression graph images alone are insufficient without temporal context from the raw time series. These results confirm that the two input representations capture complementary features: the 1D pathway captures temporal dynamics and sequential dependencies, while the 2D pathway encodes spatial distributions of local trends via the regression graph transformation. The synergy between these pathways is essential for robust CCPR performance [13].

Among the tested wavelet families (Haar, Symlet, Daubechies), the Daubechies db4 wavelet yields the highest performance (G-mean = 0.8602, F1-score = 0.8567, AUROC = 0.9312). Haar wavelets, while computationally efficient, produce lower G-mean (0.8412) due to their discontinuous basis functions, which may introduce artifacts when denoising smooth control chart features. Symlet wavelets (sym4) perform better than Haar but still lag behind Daubechies (G-mean = 0.8478 vs. 0.8602). The Daubechies db4 wavelet is well-suited for CCPR because its compact support and smooth vanishing moments effectively separate signal from noise in features extracted from control chart patterns. These findings suggest that wavelet selection should be tailored to the application domain; for CCPR, Daubechies db4 offers an optimal balance between denoising effectiveness and computational efficiency [15], [18].

Varying XGBoost complexity reveals a trade-off between performance and computational cost [22]. Reducing the number of estimators to 50 with a lower learning rate (0.05) decreases G-mean slightly to 0.8534, while increasing complexity to 200 estimators with higher learning rate (0.2) and depth (5) provides minimal additional gain (G-mean = 0.8589) but increases training time by 28% (267 s vs. 209 s). The proposed configuration (100 estimators, learning rate 0.1, max depth 3) achieves the best balance, attaining near-optimal performance (G-mean = 0.8602) with moderate computational cost. This configuration prevents overfitting on the training data while maintaining generalization to the test set, as evidenced by the high AUROC (0.9312). Training time increases modestly with model complexity: from 142 s for single-channel 1D to 209 s for the full DeepHybridCS-2CCNN. The additional computational cost of the dual-channel architecture (45 s) and wavelet denoising (22 s) is justified by the substantial perfor-

Table 10. Ablation study results for the severely imbalanced Uptrend pattern (1:200 ratio)

Configuration	Description	G-mean	F1-score	AUROC	Training Time (s)
Single-channel (1D only)	CNN with 1D time series input only, no regression graph	0.7823	0.7645	0.8734	142
	CNN with 2D regression graph input only, no 1D data	0.7456	0.7289	0.8512	156
Dual-channel (1D+2D)	CS-2CCNN with both 1D and 2D inputs (proposed)	0.8365	0.8289	0.9145	187
Dual-channel + Haar wavelet	DeepHybridCS-2CCNN with Haar wavelet denoising	0.8412	0.8334	0.9178	204
Dual-channel + Symlet wavelet	DeepHybridCS-2CCNN with Symlet (sym4) denoising	0.8478	0.8401	0.9234	206
Dual-channel + Daubechies wavelet (db4)	DeepHybridCS-2CCNN with Daubechies (db4) denoising (proposed)	0.8602	0.8567	0.9312	209
XGBoost (50 estimators, lr=0.05)	DeepHybridCS-2CCNN with reduced XGBoost complexity	0.8534	0.8489	0.9267	198
XGBoost (100 estimators, lr=0.1, depth=3)	DeepHybridCS-2CCNN with proposed XGBoost config	0.8602	0.8567	0.9312	209
XGBoost (200 estimators, lr=0.2, depth=5)	DeepHybridCS-2CCNN with increased XGBoost complexity	0.8589	0.8545	0.9298	267

mance gains (G-mean improvement of 9.95% from single-channel 1D to DeepHybridCS-2CCNN). For real-time industrial applications, a training time of approximately 3.5 minutes on standard hardware remains practical, and inference time is negligible (< 1 second per sample).

The ablation study demonstrates that each component of the proposed framework contributes meaningfully to overall performance [5]. The dual-channel architecture is essential for capturing diverse feature representations, wavelet denoising enhances feature quality by removing noise, and XGBoost classification provides robust final predictions. Practitioners seeking to adapt this work to other imbalanced time-series classification problems should prioritize the dual-channel structure and experiment with wavelet families (favoring Daubechies for smooth signals) and moderate XGBoost complexity to balance accuracy and efficiency. Training time increases modestly with model complexity, ranging from 142 s for the single-channel 1D configuration to 209 s for the full DeepHybridCS-2CCNN on an Intel i7-12900H 2.5 GHz processor with 64 GB RAM. The additional computational cost of the dual-channel architecture (45 s) and wavelet denoising (22 s) is justified by substantial performance gains (9.95% G-mean improvement from single-channel 1D to DeepHybridCS-2CCNN). Inference time remains negligible (< 1 second per sample), enabling real-time deployment in industrial quality control systems. Memory requirements are modest (< 2 GB during training), and the model scales efficiently to larger datasets through batch processing. These characteristics make DeepHybridCS-2CCNN suitable for edge deployment in smart manufacturing environments, where computational resources may be constrained but real-time abnormal pattern detection is critical [4], [6].

6. Conclusions

In this study, a new architecture is presented to address the dual challenges of data noise and class imbalance in machine learning-based CCPR, enabling more accurate and reliable predictions compared to existing methods. This work introduces two novel models: CS-2CCNN, which employs cost-sensitive learning and a dual-channel (1D + 2D) convolutional architecture to extract complementary temporal and spatial features, and DeepHybridCS-2CCNN, which enhances CS-2CCNN by integrating empirical Bayesian wavelet denoising and XGBoost classification for robust feature refinement and adaptive prediction. Experimental evaluations on simulated datasets with varying imbalance ratios (1:20 and 1:200) and the real-world Wafer dataset demonstrate the superior performance of the proposed models across multiple metrics (G-mean, F1-score, MCC, AUROC). DeepHybridCS-2CCNN achieves G-mean values exceeding 0.86 for severely imbalanced Uptrend patterns, representing a 123% improvement over the baseline cost-sensitive CNN and outperforming traditional resampling methods (SMOTE, ADASYN) by 28–32%. The ablation study (Section 4.4) confirms that each component—dual-channel architecture, wavelet denoising (Daubechies db4), and moderate XGBoost complexity—contributes meaningfully to overall performance, validating the robustness of design choices in this work. By combining advanced signal processing (wavelet denoising), adaptive classification (XGBoost), and cost-sensitive learning, this work offers a scalable and efficient solution for industrial quality control in imbalanced manufacturing environments, surpassing the accuracy and flexibility of prior models without relying on synthetic data augmentation or computationally expensive oversampling.

The superior performance of DeepHybridCS-2CCNN in detecting rare abnormal patterns has direct operational implications for modern smart manufacturing. Early detection of process deviations enables proactive interventions—such as equipment maintenance, parameter adjustments, or quality inspections—before defects propagate downstream, thereby reducing production costs, minimizing rework, and preventing costly product recalls. By achieving G-mean values exceeding 0.86 and F1-scores above 0.85 on severely imbalanced datasets, this work framework ensures that critical minority-class anomalies are identified with high reliability, even in the presence of overwhelming normal-class data. Furthermore, the cost-sensitive learning approach aligns directly with industrial risk management priorities, where the cost of missing a defective product (false negative) far exceeds the cost of unnecessary inspection (false positive). These capabilities position DeepHybridCS-2CCNN as a practical, deployable solution for quality control in sectors such as semiconductor manufacturing, automotive production, pharmaceutical processing, and other domains where rare but impactful abnormal patterns must be detected reliably and efficiently.

This research provides a solid foundation for future investigations into deeper architectures, generative models for minority-class synthesis, and extensions to multivariate and multi-class CCPR problems. The proposed approach incorporates several

complementary strategies to address class imbalance: a class-specific attention mechanism is used to emphasize features relevant to minority classes, particularly in intermediate network layers, while multi-task learning jointly performs classification and minority-class data reconstruction to improve sensitivity to sparse samples. Generative models such as GANs or VAEs are employed to synthesize realistic minority-class samples as an effective alternative to traditional over- or under sampling methods. In addition, a combined loss function integrates classification and reconstruction losses with dynamically weighted minority classes, supported by a balance evaluation module that continuously monitors performance across classes and guides parameter adjustment during training. These mechanisms can be further enhanced by complementary architectures applied after initial balancing techniques (e.g., SMOTE or ADASYN), and model performance is evaluated using a comprehensive set of metrics beyond the geometric mean, including F1-score, AUROC, Average Precision, MCC, Balanced Accuracy, Cohen's Kappa, and Log Loss.

In future studies, the development of deeper versions of CS-2CCNN models to investigate more complex abnormal patterns and apply them to datasets with a larger number of variables can be the next step in the development of this research. Also, the extension of these models to other applications based on time series data can also provide new horizons in industrial data analysis and rare event prediction.

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] D. C. Montgomery, Statistical Quality Control. New York, NY, USA: Wiley, 2009.

[2] C.-S. Cheng, Y. Ho, and T.-C. Chiu, "End-to-end control chart pattern classification using a 1D convolutional neural network and transfer learning," Processes, vol. 9, no. 9, p. 1484, 2021, doi: 10.3390/pr9091484.

[3] A. Z. Nazori, G. Brotosaputro, U. B. Solichin, and D. Mahdiana, "Design and implementation of fuzzy intelligent controller to manage energy resources in a greenhouse system," Procedia Environ. Sci. Eng. Manag., vol. 11, no. 1, pp. 19–25, 2023.

[4] Y. Al-Assaf, "Recognition of control chart patterns using multi-resolution wavelets analysis and neural networks," Comput. Ind. Eng., vol. 47, no. 1, pp. 17–29, 2004, doi: 10.1016/j.cie.2004.02.007.

[5] V. Ranaee, A. Ebrahimzadeh, and R. Ghaderi, "Application of the PSO–SVM model for recognition of control chart patterns," ISA Trans., vol. 49, no. 4, pp. 577–586, 2010, doi: 10.1016/j.isatra.2010.06.005.

[6] P. Xanthopoulos and T. Razzaghi, "A weighted support vector machine method for control chart pattern recognition," Comput. Ind. Eng., vol. 70, pp. 134–149, 2014, doi: 10.1016/j.cie.2014.01.014.

[7] D. Fuqua and T. Razzaghi, "A cost-sensitive convolution neural network learning for control chart pattern recognition," Expert Syst. Appl., vol. 150, p. 113275, 2020, doi: 10.1016/j.eswa.2020.113275.

[8] X. Zhou, P. Jiang, and X. Wang, "Recognition of control chart patterns using fuzzy SVM with a hybrid kernel function," J. Intell. Manuf., vol. 29, pp. 51–67, 2018, doi: 10.1007/s10845-015-1089-6.

[9] Y. Yu and M. Zhang, "Control chart recognition based on the parallel model of CNN and LSTM with GA optimization," Expert Syst. Appl., vol. 185, p. 115689, 2021, doi: 10.1016/j.eswa.2021.115689.

[10] A. Maged and M. Xie, "Recognition of abnormal patterns in industrial processes with variable window size via convolutional neural networks and AdaBoost," J. Intell. Manuf., vol. 34, no. 4, pp. 1941–1963, 2023, doi: 10.1007/s10845-021-01907-8.

[11] X. Zhang, H. Peng, J. Zhang, and Y. Wang, "A cost-sensitive attention temporal convolutional network based on adaptive top-k differential evolution for imbalanced time-series classification," Expert Syst. Appl., vol. 213, p. 119073, 2023, doi: 10.1016/j.eswa.2022.119073.

[12] Y. Gao, L. Gao, X. Li, and S. Cao, "A hierarchical training-convolutional neural network for imbalanced fault diagnosis in complex equipment," IEEE Trans. Ind. Informatics, vol. 18, no. 11, pp. 8138–8145, 2022, doi: 10.1109/TII.2022.3160420.

[13] L. Xue, H. Wu, H. Zheng, and Z. He, "Control chart pattern recognition for imbalanced data based on multi-feature fusion using convolutional neural network," Comput. Ind. Eng., vol. 182, p. 109410, 2023, doi: 10.1016/j.cie.2023.109410.

[14] J. He, L. Yin, J. Liu, C. Zhang, and H. Yang, "A fault diagnosis method for unbalanced data based on a deep cost sensitive convolutional neural network," IFAC-PapersOnLine, vol. 55, no. 3, pp. 43–48, 2022, doi: 10.1016/j.ifacol.2022.05.008.

[15] D. Dablain, K. N. Jacobson, C. Bellinger, M. Roberts, and N. V. Chawla, "Understanding CNN fragility when learning with imbalanced data," Mach. Learn., vol. 113, no. 7, pp. 4785–4810, 2024, doi: 10.1007/s10994-023-06326-9.

[16] C.-S. Cheng, P.-W. Chen, Y.-C. Hsieh, and Y.-T. Wu, "Multivariate process control chart pattern classification using multi-channel deep convolutional neural networks," Mathematics, vol. 11, no. 15, p. 3291, 2023, doi: 10.3390/math11153291.

[17] Farahnakian et al., "Addressing imbalanced data for machine learning based mineral prospectivity mapping," Ore Geol. Rev., p. 106270, 2024, doi: 10.1016/j.oregeorev.2024.106270.

[18] T. Hong, Y. Yu, and J. Chen, "Research on the method of online abnormal patterns recognition in the control chart of smart meter based on ML-SVM," SSRN Electron. J., preprint, 2023, doi: 10.2139/ssrn.4403225.

[19] D. Pham and E. Oztemel, "Combining multi-layer perceptrons with heuristics for reliable control chart pattern classification," WIT Trans. Inf. Commun. Technol., vol. 1, pp. 801-810, 1993.

[20] K. Zhou, Y. Chen, W. Xiong, J. Zhang, and X. Gong, "Control chart pattern recognition for small samples based

on Siamese neural network," Qual. Eng., vol. 37, no. 1, pp. 64–78, 2025, doi: 10.1080/08982112.2024.2340529.
[21] T. Zan, X. Jia, X. Guo, M. Wang, X. Gao, and P. Gao, "Research on variable-length control chart pattern recognition based on sliding window method and SECNN-BiLSTM," Sci. Rep., vol. 15, no. 1, p. 5921, 2025, doi: 10.1038/s41598-025-86849-4.
[22] A. A. Khan, O. Chaudhari, and R. Chandra, "A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation," Expert Syst. Appl., vol. 244, p. 122778, 2024, doi: 10.1016/j.eswa.2023.122778.
[23] J. Xie, L. Sun, and Y. F. Zhao, "On the data quality and imbalance in machine learning-based design and manufacturing—A systematic review," Engineering, vol. 45, pp. 105–131, 2025, doi: 10.1016/j.eng.2024.04.024.
[24] A. Ostonokulov, D. Sholdarov, A. Kuliboyev, N. Yuldasheva, D. Abdujalilova, and G. Adilova, "The impact of education management on entrepreneurship development and emerging markets," Econ. Ann.-XXI, vol. 211, no. 9–10, pp. 40–45, 2024, doi: 10.21003/ea.V211-06.
[25] R. Adattil, P. Thorvald, and D. Romero, "Assessing the psychosocial impacts of Industry 4.0 technologies adoption in the Operator 4.0: Literature review and theoretical framework," Int. J. Ind. Eng. Manag., vol. 15, no. 1, pp. 59–80, 2024, doi: 10.24867/IJIEM-2024-1-348.
[26] M. A. Baharuddin and I. Masood, "A study in development of control chart pattern recognition scheme based on literature," Res. Prog. Mech. Manuf. Eng., vol. 3, no. 1, pp. 10–25, 2022.
[27] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel and R. Togneri, "Cost-Sensitive Learning of Deep Feature Representations From Imbalanced Data," IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 8, pp. 3573-3587, 2018, doi: 10.1109/TNNLS.2017.2732482.
[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Commun. ACM, vol. 60, no. 6, pp. 84–90, Jun. 2017, doi: 10.1145/3065386.
[29] I. M. Johnstone and B. W. Silverman, "Needles and straw in haystacks: Empirical Bayes estimates of possibly sparse sequences," Ann. Stat., vol. 32, no. 4, pp. 1594–1649, 2004, doi: 10.1214/009053604000000030.

Appendix 1

Table A. A quick survey of the literature review

Authors	Methodology	Key Contribution	Strengths	Limitations	Addresses Imbalanced Data	Real-World Applicability
[3]	Fuzzy Systems	Early use of fuzzy logic for CCPR	Simple, interpretable	Limited scalability for large, noisy datasets	No	Low
[4]	Artificial Neural Networks (ANNs)	Applied ANNs for pattern recognition	Effective for small datasets	Struggles with imbalanced and turbulent data	No	Moderate
[5]	Support Vector Machines (SVM)	SVM-based CCPR with good accuracy	Robust for simpler datasets	Poor performance with high-dimensional, imbalanced data	No	Moderate
[6]	Weighted SVM	Introduced weighted SVM to handle data turbulence	Improved handling of noisy data	Limited focus on class imbalance	Partial	Moderate
[8]	Convolutional Neural Networks (CNN)	Achieved 99.56% accuracy on small datasets	High accuracy, effective feature extraction	Limited to specific defect types	No	High
[7]	Cost-Sensitive CNN	Pioneered cost-sensitive CNN for imbalanced time-series data	High accuracy, flexible for imbalanced data	Limited generalizability across diverse processes	Yes	High
[9]	CNN + LSTM	Hybrid CNN-LSTM model for time-series classification	Superior performance in sequential data	Computationally intensive	No	High
[12]	Hierarchical CNN	Detected faults in complex equipment with 96.56% accuracy	Effective for complex systems	Limited focus on real-time adaptability	Yes	High
[24]	Systematic Review	Highlighted lack of research in CCPR	Comprehensive overview	No new methodology	No	N/A

Authors	Methodology	Key Contribution	Strengths	Limitations	Addresses Imbalanced Data	Real-World Applicability
[10]	CNN + AdaBoost	Achieved 99.78% detection accuracy	High accuracy, robust ensemble approach	Limited to specific datasets	No	High
[11]	Cost-Sensitive CNN with Attention	Improved G-mean for imbalanced time-series classification	Robust to class imbalance	Complex architecture, high computational cost	Yes	High
[18]	Online Detection Algorithm	95.3% accuracy for smart meter control charts	Real-time applicability	Lower accuracy compared to CNN-based methods	No	High
[16]	MCDCNN Model	10% improvement in multivariate pattern classification	Tailored for multivariate data	Limited focus on real-time processing	No	High
[19]	Hybrid System (Rule-Based + MLP)	Real-time anomaly detection with modular design	Flexible, scalable	Limited evaluation across diverse datasets	No	High
[23]	Active learning to mitigate issues like data scarcity	Providing a robust framework supported by 94 analyzed publications	Categorization of data terminologies and challenges	Lack of detailed quantitative comparisons and limited focus	Yes	No
[22]	Review Paper	Reviews ensemble learning & data augmentation techniques	Robust performance, with random forests and XGBoost	High computational costs	Yes	No
[17]	Combination of integration of some techniques into QGIS	Using QGIS	produces geologically sensible prospectively maps	Complexity and significant computational resources	Yes	Yes
[20]	PCSNN (Siamese Neural Network)	Excelled in small-sample pattern recognition	Effective for limited data	Limited scalability for large datasets	No	Moderate
[21]	SECNN-BiLSTM	High accuracy for variable-length pattern recognition	Handles variable data effectively	Complex preprocessing requirements	No	High
Current Study	DeepHybridCS-2CCNN (CS-2CCNN + Wavelet Denoising + XGBoost)	Multi-channel CNN with wavelet denoising and XGBoost for robust CCPR	Addresses imbalance, noise, and dynamic patterns; high accuracy	Requires further validation across diverse industries	Yes	High

Appendix 2: Technical Details of Regression Graph Transformation and Wavelet Denoising

A2.1 Regression Graph Transformation for 2D Image Construction

The regression graph transformation converts univariate time series data into two-dimensional gray-scale images, enabling the dual-channel architecture to capture spatial features alongside temporal patterns. This technique, adapted from Cheng et al. [16], proceeds as follows.

Given a time series $X=[x_1,x_2,\ldots,x_T]$ of length $T=20$, we construct a regression graph by fitting a piecewise linear model to consecutive data points. For each pair of consecutive points (t_i,x_i) and (t_{i+1},x_{i+1}), we compute the local slope $m_i=\frac{x_{i+1}-x_i}{t_{i+1}-t_i}$ and intercept $c_i=x_i-m_i\cdot t_i$. These local regression parameters are then discretized into a fixed-size grid (e.g., pixels) where each cell's intensity represents the density of

regression lines passing through that region of the slope-intercept space.

Specifically, the transformation maps the slope m_i to the horizontal axis and the intercept c_i to the vertical axis of the image. The intensity of pixel (u,v) in the output image is computed as $I(u,v)=\sum_{i=1}^{T-1}\mathbf{1}\left(|m_i-\mu(u)|\le\delta_m\right),\ \mathbf{1}\left(|c_i-\gamma(v)|\le\delta_c\right)$, where $\mu(u)$ and $\gamma(u)$ are the slope and intercept values corresponding to pixel coordinates (u,v), and δ_m, δ_c are tolerance thresholds. This process creates a grayscale image that encodes the distribution of local trends within the time series. The resulting 20×20 regression graph images serve as input to the 2D convolutional pathway of CS-2CCNN and DeepHybridCS-2CCNN, complementing the raw 1D time series input.

The regression graph transformation is threshold-free (unlike traditional Gramian Angular Field or Markov Transition Field methods), making it robust to variations in data scale and reducing sensitivity to outliers. This property is particularly valuable for control chart data, where abnormal patterns may exhibit varying magnitudes.

A2.2 Empirical Bayesian Wavelet Denoising Procedure

Wavelet denoising enhances the quality of features extracted by CS-2CCNN before they are fed into the XGBoost classifier in DeepHybridCS-2CCNN. This study employs the empirical Bayesian wavelet denoising method proposed by Johnstone and Silverman [29], which adaptively thresholds wavelet coefficients based on a data-driven prior distribution. The wavelet denoising procedure consists of four steps:

1. Wavelet Decomposition: The feature vector $\mathbf{f}\in\mathbf{w}^d$ extracted from the concatenated CNN layers undergoes a discrete wavelet transform (DWT) using the Daubechies wavelet family (db4), yielding wavelet coefficients $\mathbf{w}=[x_1,x_2,\ldots,w_d]$ at multiple resolution levels.

2. Empirical Bayes Threshold Estimation: For each resolution level j, the empirical Bayesian method models the distribution of wavelet coefficients as a mixture of a point mass at zero (representing noise) and a heavy-tailed distribution (representing signal). The threshold τ_j for level j is estimated by maximizing the marginal likelihood of the observed coefficients under this mixture model. This adaptive threshold balances false positives (removing signal) and false negatives (retaining noise).

3. Soft Thresholding: Wavelet coefficients are thresholded using the soft-thresholding operator $\tilde{w}_i=\text{sign}(w_i)\cdot\max(|w_i|-\tau_j,0)$ where τ_j is the empirical Bayes threshold for the resolution level containing coefficient w_i. Soft thresholding shrinks coefficients toward zero, preserving continuity and reducing artifacts compared to hard thresholding.

4. Inverse Wavelet Transform: The denoised wavelet coefficients $\mathbf{w}$ are transformed back to the feature domain via the inverse DWT, yielding the denoised feature vector $\tilde{\mathbf{f}}$, which is then passed to the XGBoost classifier.

The empirical Bayesian approach is advantageous because it does not require manual tuning of the threshold parameter; instead, the threshold is derived from the data itself, adapting to the noise characteristics of each feature set. This adaptability is critical in DeepHybridCS-2CCNN, where feature distributions vary across different control chart patterns and imbalance ratios.

In this study, wavelet denoising is applied to the 140-dimensional feature vector (concatenated from the 1D and 2D pathways) extracted after the global average pooling layers of CS-2CCNN. Empirical results (see Section 4.4, Ablation Study) demonstrate that wavelet denoising improves G-mean by 2–4% across most patterns, confirming its value in enhancing feature quality for downstream classification.