Towards Reliable Recognition of Concurrent Abnormal Patterns in Control Charts Using Multi-Label Deep Learning

Mohammed Modar[1], Abdelilah Ganmati[2], Omar El Farissi[3]
Industrial Engineering Department, National School of Applied Sciences, Agadir, Morocco[1]
Laboratory of Computer Systems and Vision (LABSIV), Faculty of Sciences, Agadir, Morocco[2]
Mechanical Engineering Department, National School of Applied Sciences, Agadir, Morocco[3]

***Abstract*—Control charts do more than raise an alarm: their shapes can give an early indication of what has changed in a process. This study considers the case in which one chart window contains more than one abnormal behavior. The observed sequence is then a mixture rather than a pure pattern. We formulate this problem directly as multi-label classification. A one-dimensional CNN receives a raw-scale window of 32 observations and predicts the active elementary labels. The controlled protocol contains twelve scenarios: normal behavior, six single abnormal patterns, and five selected concurrent patterns. Raw-scale input is retained because shift patterns depend partly on level information that may be weakened by window-wise normalization. The retained training setup gives additional exposure to difficult shift and trend cases, while validation and testing remain balanced. Across five repeated trainings, the model achieved 96.11% exact match accuracy, 96.41% precision, 96.46% recall, 96.44% F1-score, and 1.04% Hamming loss. The 95% confidence interval for exact match was 96.05–96.17%. Additional analyses show stable performance around a decision threshold of 0.5, strong cyclic and systematic recognition, and lower performance for short shift cases. The results support direct multi-label CNN recognition for the selected protocol. Broader shift-containing mixtures, more complex combinations, varying noise conditions, and real industrial validation remain outside the scope of the present controlled study.**

Keywords*—Control chart pattern recognition; concurrent patterns; statistical process control; multi-label classification; convolutional neural network; process monitoring

I. Introduction

Control charts remain one of the basic tools of statistical process control. When a process is stable, the plotted observations fluctuate around a reference level without a persistent structure. When an assignable cause appears, the chart may drift, shift to another level, oscillate, or alternate in a systematic way [1]–[5]. These shapes are useful because they can provide an early diagnostic indication. A trend may be associated with progressive tool wear or a gradual operating change, while a shift may reflect an abrupt change in machine settings, material, or method. Cyclic and systematic patterns may indicate periodic influences or alternating production conditions [1], [5], [16], [23].

The goal of control chart pattern recognition (CCPR) is to identify the abnormal behavior visible in a chart window. Earlier studies used neural networks, statistical features, support vector machines, and optimization-based classifiers [6]–[13]. More recent work has used convolutional models, recurrent networks, feature fusion, transfer learning, and attention-based architectures [23], [29], [36]–[39].

Much of the CCPR literature assumes that one window contains one dominant abnormal pattern. This assumption is useful for basic classification, but it is restrictive in practice. Two assignable causes may act at the same time, so a chart can drift while it oscillates, or show a systematic alternation while another change is developing.

Concurrent CCPR addresses this case. It is difficult because the components are mixed in the same short and noisy sequence. A local increase may be random variation, the start of a trend, or part of a shift. A periodic movement may also be partly hidden by noise. The model should therefore indicate which elementary components are present rather than only select the nearest exclusive class.

Several concurrent CCPR studies transform the signal before classification. Wavelet decomposition, independent component analysis, singular spectrum analysis, blind source separation, learning vector quantization, support vector machines, random forests, and extreme learning machines have been used for this purpose [14]–[22]. These studies show that concurrent CCPR is not a minor extension of ordinary pattern recognition. They also show that the representation matters: when decomposition is inaccurate, its error is passed to the classifier.

Another issue is the representation of the target. A common approach is to create a separate class for each mixture. This is manageable when the number of combinations is small, but every new mixture requires a new class and the elementary diagnostic components remain hidden inside the class name. In this work, each elementary behavior has its own output. A window containing an upward trend and a cyclic movement should activate both labels. This description is closer to the way an engineer reads the chart.

The present study therefore focuses on a direct multi-label formulation. We use a compact 1D CNN, independent sigmoid outputs, and multi-label evaluation metrics. We keep the input on its original scale because shifts are partly defined by a level change. The main protocol contains twelve scenarios: normal behavior, six single abnormal patterns, and five concurrent cases. This set is deliberately focused. It is large enough to test direct concurrent recognition while remaining clear enough for

pattern-level error analysis.

We also compare a small number of reasonable training variants under the same protocol. The aim is not to claim a universal architecture or an exhaustive hyperparameter search. It is to establish a transparent reference configuration and examine its stability, label-wise behavior, threshold sensitivity, computational cost, and practical scope.

The main contributions are as follows:

- Concurrent CCPR is formulated as a direct multi-label problem so that the elementary abnormal components remain visible in the output.
- A twelve-scenario protocol is defined with explicit encoding for normal, single abnormal, and selected concurrent cases.
- A compact 1D CNN is evaluated on raw-scale windows using full-vector and label-wise metrics across repeated runs.
- Additional analyses examine uncertainty, threshold sensitivity, label-wise errors, computational cost, and the practical limits of the focused experimental scope.

The remainder of the study is organized as follows. Section II reviews related CCPR and multi-label studies. Section III presents the methodology. Section IV describes the experimental setup. Section V reports and discusses the results. Section VI concludes the study.

II. Literature Review

A. Single Control Chart Pattern Recognition

Early CCPR studies generally considered one abnormal pattern at a time. Neural networks were used to map a chart window to a pattern class [6], [7], while other studies used statistical or shape-based features to improve robustness to noise [10]–[12]. Support vector machines and optimization-based models were later adopted for nonlinear decision boundaries [13]. More recent work has introduced 1D CNNs, feature fusion, transfer learning, recurrent networks, and methods designed for imbalanced data or small shifts [29], [36]–[39].

The single-pattern setting remains useful because it isolates the shapes of trends, shifts, cycles, and systematic alternations. Its main limitation is that each window is forced into one class even when more than one source is active.

B. Concurrent Pattern Recognition

Concurrent CCPR was developed for mixed abnormal structures. Table I summarizes representative approaches in this area. Many methods first transform or decompose the signal and then classify the resulting representation. Wavelet methods use time-frequency information [14], [17]; ICA and blind source separation attempt to recover hidden components [15], [20], [21]; and singular spectrum analysis has been combined with SVM, LVQ, and random forest classifiers [16], [18]. Other studies have used ESMD, extreme learning machines, and PCA-weighted strategies [19], [22].

These methods show two recurring issues. First, overlapping patterns are harder to separate than single patterns. Second, performance depends not only on the classifier but also on the representation produced before classification. A decomposition stage can be useful, but it also introduces another source of error.

The target representation is equally important. Composite-class methods assign one class to each predefined mixture. A label-wise formulation instead keeps the same elementary labels across single and concurrent windows. For example, the cyclic label has the same meaning whether it appears alone, with a trend, or with another component. This representation supports a more direct diagnostic reading, although it does not by itself identify the physical root cause.

C. Deep Learning and Multi-Label Learning

Deep learning is now widely used for time-series classification and process monitoring [25]–[27]. For short fixed-length windows, a 1D CNN is a natural baseline because its filters can react to local rises, drops, transitions, oscillations, and alternating segments [28], [29], [35], [36]. Recurrent, hybrid CNN–LSTM, residual, and attention-based models are useful alternatives when longer temporal dependencies or variable-length sequences are central [24], [38], [39].

The output layer is central to the present problem. Softmax is appropriate when only one class can be active. Concurrent patterns require independent outputs because a cyclic component can coexist with a trend or a shift. We therefore use one sigmoid output per label and optimize binary cross-entropy, following standard multi-label learning practice [32]–[34].

The evaluation must also reflect partial correctness. Exact match requires the entire label vector to be correct, whereas Hamming loss counts individual label errors. Precision, recall, and F1-score show whether the model tends to create false activations or miss active components. The present work combines concurrent CCPR with this label-wise output and evaluation view.

III. Proposed Methodology

Fig. 1 shows the workflow we used in the experiments. We start from basic control chart components and use them to create single-pattern windows and selected concurrent windows. After that, each window is encoded as a vector of seven binary labels and passed to a one-dimensional CNN. We then read the prediction as a multi-label output, not as a single class.

We kept the method simple for a reason. If the generator is not clear, it becomes difficult to know why the model succeeds or fails. Also, the target must follow the physical meaning of the signal. If two components are used to generate the chart, two labels should be active. Finally, the evaluation should not hide partial mistakes. Detecting one component and missing the second is not the same error as detecting nothing.

This also explains why we do not try to solve all possible CCPR cases at once. The number of possible concurrent combinations grows quickly when more basic patterns are added. A very large set would look more complete, but it would be harder to understand. Here, the aim is to keep enough cases to test the concurrent idea and still be able to read the errors pattern by pattern.

TABLE I. REPRESENTATIVE CHOICES IN CONCURRENT CCPR LITERATURE

Study	Main strategy	Output view	Main relevance to the present work
Al-Assaf [14]	Wavelet-based multiresolution analysis	Pattern class	Shows that time-frequency representation can support concurrent CCPR.
Lu et al. [15], [21]	ICA combined with SVM	Separated patterns	Uses decomposition before classification; performance depends on separation quality.
Xie et al. [16]	SSA combined with SVM	Composite class	Represents concurrent windows through extracted temporal components.
Du et al. [17]	Wavelet decomposition and multiclass SVM	Composite class	Treats each concurrent case as a class after transformed feature extraction.
Gu et al. [18]	SSA and learning vector quantization	Composite class	Uses decomposition and prototype-based recognition for mixed patterns.
Yang et al. [19]	ESMD and extreme learning machine	Components	Addresses identification and quantification of concurrent patterns.
Pelegrina et al. [20]	Blind source separation and feature extraction	Separated sources	Highlights the importance and sensitivity of signal separation.
Cheng et al. [23]	Multi-label 1D CNN	Active labels	Provides the closest direct multi-label CNN reference for concurrent CCPR.

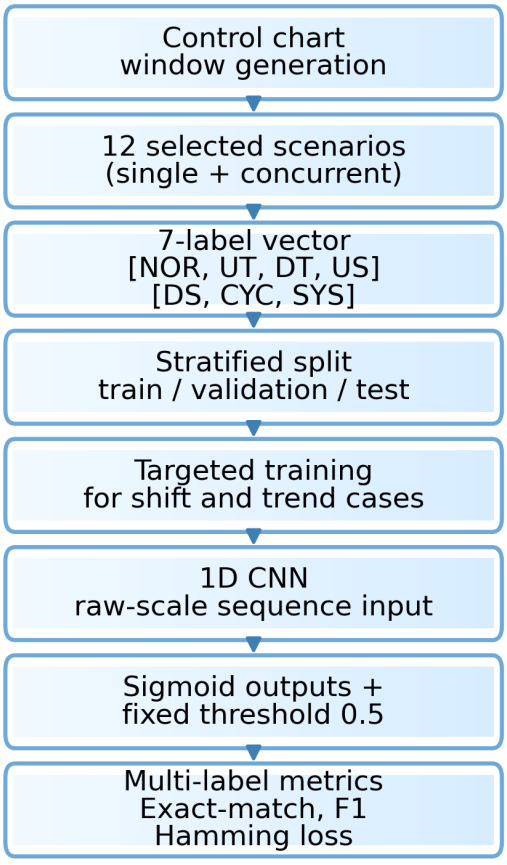

Fig. 1. Experimental workflow used in the proposed study.

We use the same logic for the model. We do not add a decomposition block or a long list of handcrafted features. Such elements may be useful later, but they would make it harder to see the effect of the multi-label formulation itself. We therefore use the present design as a direct reference point: raw window, CNN, sigmoid labels, and multi-label metrics.

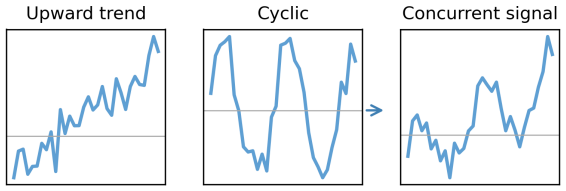

Fig. 2. Illustration of the transition from single abnormal patterns to a concurrent observed signal.

A. From Single Pattern to Concurrent Patterns

Going from single patterns to concurrent patterns changes the problem quite strongly. In the single case, the model searches for one dominant abnormal source. In the concurrent case, two sources may be present at the same time. The target is therefore not a class name anymore; it is the set of active components.

Fig. 2 gives a small example. One signal has a trend, another has a cyclic movement, and the final signal is the visible mixture. A classical multiclass model would need a special class for this mixture. With the multi-label view, the two components remain visible. This is more useful for diagnosis, because the output says what is inside the mixture.

B. Problem Formulation

Let a control chart window be represented as a one-dimensional sequence:

$$\mathbf{x} = [x_1, x_2, \ldots, x_n]^T, \tag{1}$$

where, $n = 32$ in our study. We aim to learn a function f_θ that maps the input window to a vector of label probabilities:

$$\mathbf{p} = f_\theta(\mathbf{x}) = [p_{NOR}, p_{UT}, p_{DT}, p_{US}, p_{DS}, p_{CYC}, p_{SYS}], \tag{2}$$

where, each element belongs to $[0, 1]$. The corresponding ground-truth vector is:

$$\mathbf{y} = [y_1, y_2, \ldots, y_7], \quad y_k \in \{0, 1\}. \tag{3}$$

We represent a concurrent pattern by several active labels in $\mathbf{y}$. For example, an upward trend with a cyclic component has $y_{UT} = 1$ and $y_{CYC} = 1$.

We do not treat this as a softmax problem. Softmax makes the classes compete, which is correct only when one class excludes the others. Here, one label does not exclude another one. A cyclic component can appear with a trend. For that reason, each output neuron uses a sigmoid activation.

We keep the normal label as an explicit output. We do not combine it with abnormal labels in our generated scenarios. A normal window means that no deterministic abnormal component has been added. This gives the CNN a reference behavior, against which the abnormal components are learned.

We make this point to avoid a small ambiguity. Random noise is present in all windows, but the label NOR does not mean 'noise'. It means that none of the deterministic abnormal components of the protocol is present.

C. Pattern Set and Label Encoding

We use twelve scenarios. Seven are single-label cases: normal, upward trend, downward trend, upward shift, downward shift, cyclic, and systematic. The remaining five are concurrent cases built from trend, cyclic, and systematic components. Shift-containing mixtures are not part of the present concurrent set. This choice is intentional rather than a conceptual exclusion. In a short window of 32 observations, the visible form of a shift depends jointly on its magnitude, change-point location, and the number of post-change observations. When a shift is superposed on another component, the relative amplitudes and temporal alignment introduce additional factors that require a dedicated generation and training protocol. Adding these cases to the present experiment would therefore change the controlled scope rather than simply add another class.

The scenario set is consequently a focused reference rather than an exhaustive catalogue of industrial mixtures. It was selected to examine the direct multi-label formulation under clearly defined and interpretable conditions. We do not claim that the resulting model covers every possible combination. We also do not use scenario names as model outputs. Instead, we use elementary diagnostic components. For example, *upward trend + cyclic* activates two labels, while *cyclic* activates one. Table II separates the scenario used to generate the window from the labels predicted by the model. This distinction allows a concurrent window to be described through more than one component.

Fig. 3 illustrates representative windows for the twelve scenarios used in the controlled protocol.

D. Data Generation Protocol

Each window is generated by adding a deterministic pattern component to random noise:

TABLE II. PATTERN SCENARIOS AND ACTIVE LABEL VECTORS

Scenario	Active labels	Label-vector interpretation
Normal	NOR	In-control behavior.
Upward trend	UT	Increasing slope.
Downward trend	DT	Decreasing slope.
Upward shift	US	Positive level change.
Downward shift	DS	Negative level change.
Cyclic	CYC	Oscillatory component.
Systematic	SYS	Alternating high-low structure.
Upward trend + cyclic	UT, CYC	Trend and oscillation.
Downward trend + cyclic	DT, CYC	Negative trend and oscillation.
Upward trend + systematic	UT, SYS	Trend with alternation.
Downward trend + systematic	DT, SYS	Negative trend with alternation.
Systematic + cyclic	SYS, CYC	Alternation and oscillation.

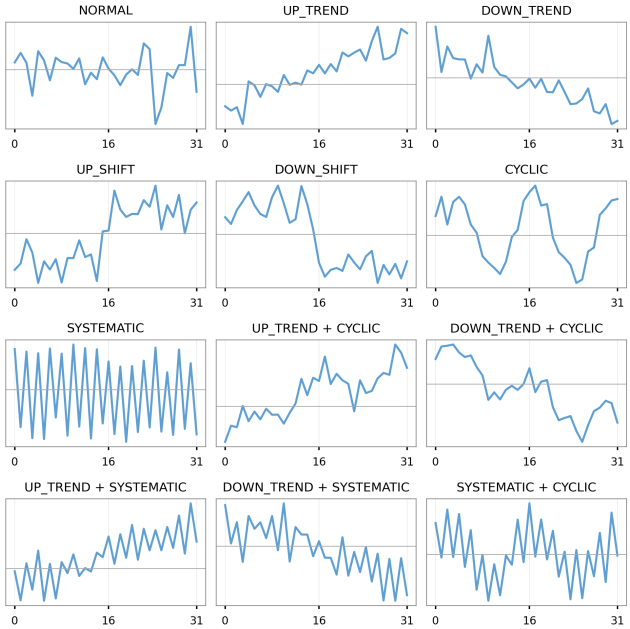

Fig. 3. Examples of the twelve generated control chart scenarios.

$$x_t = \epsilon_t + D_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2), \quad t = 1, \ldots, n. \tag{4}$$

We define the term D_t according to the active labels. The deterministic components are summarized in Table III. In a concurrent case, the deterministic components are added first, and noise is then added. In the main protocol, the Gaussian noise level is fixed at $\sigma = 1$ for all training, validation, and test windows. This controlled setting isolates the effect of the multi-label formulation and keeps the comparison between model configurations consistent. A trend creates drift, a shift creates a level change after a point, a cyclic term creates oscillation, and the systematic term gives a high-low alternation. We therefore know the labels from the generator; we do not assign them by visual judgment.

TABLE III. DETERMINISTIC COMPONENTS USED IN THE GENERATOR

Component	Expression	Role
Trend	$\pm gt$	Models progressive drift.
Shift	$\pm s\mathbb{1}(t > \tau)$	Models a sudden level change.
Cyclic	$a\sin(2\pi t/T + \phi)$	Models periodic behavior.
Systematic	$d(-1)^t$	Models alternating high-low observations.
Concurrent case	$D_t = D_t^{(1)} + D_t^{(2)}$	Adds active components in one window.

We keep the additive form plain by design. It keeps a visible link between the signal and the label vector. If the model misses a label, we know which component was missed. This is useful in a first controlled study.

In the main experiment, we keep the signal on its original scale. We do not apply window-wise z-score normalization. The reason is simple: a shift is partly a level change, and this level information can be weakened by normalizing every window separately.

We do not mean that normalization is never useful in CCPR. We only mean that, for the present target labels, preserving the vertical level is more consistent. This is especially true for upward and downward shifts.

This also makes our error discussion easier. If a shift is confused with a trend, we can still look at the original level and slope of the window. With independent normalization, part of this information has already been changed.

We also use the generator to improve reproducibility. Concurrent-CCPR papers are difficult to compare when equations, parameter ranges, or window sizes are not given in enough detail. Here, training, validation, and test windows follow the same definitions, but they are generated with different random seeds. The test data remain unseen, although they come from the same controlled process.

E. CNN Architecture

The CNN receives a raw sequence of 32 observations. Fig. 4 and Table IV summarize the retained architecture, which contains two one-dimensional convolutional layers, max pooling, dropout, one dense layer, and seven sigmoid outputs. No handcrafted descriptors or decomposition stage is added before the network.

The compact architecture is intentional. With only 32 input points, two convolutional stages are sufficient to build local features at two levels without introducing a large parameter count. The first layer detects short transitions and repeated local movements, while the second combines them into broader pattern evidence. A deeper residual network, a CNN–LSTM hybrid, or an attention mechanism may be useful for longer, variable-length, or more irregular signals [38], [39]; however, such models would add capacity and tuning choices that are outside the focused comparison made here.

The retained model has 15,687 trainable parameters and approximately 54,680 multiply-accumulate operations per sample. This small computational footprint supports its use as a reproducible reference architecture.

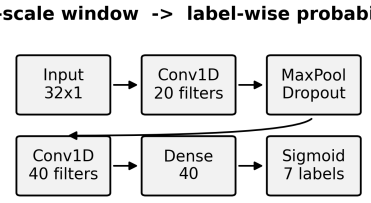

Fig. 4. CNN architecture used for multi-label recognition of control chart windows.

TABLE IV. CNN CONFIGURATION USED IN THE RETAINED EXPERIMENT

Element	Setting
Input size	32×1 raw-scale signal.
Convolution 1	20 filters, kernel size 5, ReLU.
Pooling/dropout	Max pooling followed by 12% dropout.
Convolution 2	40 filters, kernel size 3, ReLU.
Dense layer	40 units, ReLU, 12% dropout.
Output layer	Seven sigmoid neurons.
Loss	Binary cross-entropy.
Optimizer	Adam, learning rate 0.001.
Decision rule	Fixed threshold of 0.5.
Trainable parameters	15,687.
Approximate MACs	54,680 per sample.

F. Learning Objective and Decision Rule

We train the CNN using binary cross-entropy averaged over the $K = 7$ labels:

$$\mathcal{L}(\mathbf{y}, \mathbf{p}) = -\frac{1}{K}\sum_{k=1}^{K}\left[y_k \log(p_k) + (1 - y_k)\log(1 - p_k)\right]. \quad (5)$$

We interpret the sigmoid output p_k as the probability that label k is active. At inference time, the decision rule is:

$$\hat{y}_k = \begin{cases} 1, & p_k \geq 0.5, \\ 0, & p_k < 0.5. \end{cases} \quad (6)$$

We fix the threshold before testing. A label is active when its sigmoid value is at least 0.5. Label-specific thresholds could improve particular cases, but they would add another tuning stage. We therefore keep one common threshold and examine its sensitivity separately in Section V using saved prediction probabilities.

G. Evaluation Metrics

We choose the metrics according to the output. In many concurrent-CCPR studies, recognition accuracy or correct classification rate is enough because the output is one composite class [15]–[22]. Here, the output is a vector. We therefore need to look at both the full vector and the individual label decisions.

We use exact match as the strict measure: the complete vector must be correct. We use Hamming loss as a softer measure; it counts the wrong labels one by one. Precision tells

TABLE V. MULTI-LABEL METRICS RETAINED FOR EVALUATION

Metric	Expression	Interpretation
Exact match	$\frac{1}{N}\sum_i \mathbb{1}(\mathbf{y}_i = \hat{\mathbf{y}}_i)$	Strict full-vector correctness.
Precision	$TP/(TP+FP)$	Penalizes false label activations.
Recall	$TP/(TP+FN)$	Penalizes missed active labels.
F1-score	$2PR/(P+R)$	Balances precision and recall.
Hamming loss	$\frac{1}{NK}\sum_{i,k} \mathbb{1}(y_{ik} \neq \hat{y}_{ik})$	Label-wise error rate.

us whether the model activates labels too easily. Recall tells us whether it misses active labels. F1-score gives a balance. Table V summarizes these metrics and their interpretation [32]–[34].

H. Rationale for the Main Design Choices

We fixed a few choices before the final comparison. The first one was the window length. The window should be long enough to show a trend, a shift, or a cycle. It should also be short enough to keep the detection useful. We used 32 observations.

Our second choice was the raw input. Many concurrent-CCPR methods transform the signal before classification. This can help, but it also adds parameters and possible errors. In this study, we ask a narrower question: what can be achieved if the CNN sees only the observed chart window? This gives us a simple reference for later comparisons.

Our third choice was about balance. We gave each scenario the same number of test windows. The label distribution is not perfectly balanced, because some labels appear in more than one scenario. This is expected. We give the retained training setup more examples of difficult shift and trend cases, but we keep validation and testing balanced.

Our fourth choice concerned the metrics. Exact match tells us whether the whole diagnosis is correct, but it does not tell us how large the error is. For that reason, we also report Hamming loss, precision, recall, and F1-score.

Finally, we keep the number of variants limited. This is not a broad hyperparameter search. We compare a few reasonable choices: baseline CNN, two-channel input, weighted loss, shift-extra-only training, and targeted training. We keep the comparison readable.

IV. EXPERIMENTAL SETUP

The data follow the protocol described in Section III, and Table VI summarizes the main experimental settings. For each scenario, 5000 windows are generated for training and 1250 for testing. Ten percent of the training data are reserved for stratified validation. The test set remains balanced across the twelve scenarios and is not used to select the decision threshold.

The CNN is implemented in TensorFlow [30] and trained with Adam, a batch size of 16, and a maximum of 100 epochs. Validation loss is used for early stopping, and the evaluation metrics are computed with scikit-learn [31]. The experiment is repeated five times with different seeds. The retained configuration receives additional training examples for difficult shift and trend cases, but the validation and test sets remain unchanged.

TABLE VI. EXPERIMENTAL SETTINGS

Item	Value
Window size	32 observations.
Pattern scenarios	12 in the main protocol.
Output labels	7: NOR, UT, DT, US, DS, CYC, SYS.
Training samples	5000 windows per scenario before reinforcement.
Test samples	1250 windows per scenario, balanced.
Validation split	10% stratified validation.
Preprocessing	Raw-scale signal; no window-wise z-score.
Gaussian noise	Fixed at $\sigma = 1$ in all data splits.
Decision threshold	0.5 for all labels.
Number of runs	5 repeated trainings.

TABLE VII. OVERALL PERFORMANCE OF THE RETAINED CONFIGURATION

Metric	Mean (%)	Std.	95% CI (%)
Exact match	96.11	0.05	[96.05, 96.17]
Precision	96.41	0.05	[96.35, 96.47]
Recall	96.46	0.07	[96.37, 96.55]
F1-score	96.44	0.06	[96.37, 96.51]
Hamming loss	1.04	0.02	[1.02, 1.06]

No handcrafted descriptors, wavelet transform, SSA, ICA, or blind-source-separation stage is used. The main purpose is to examine how far a direct multi-label CNN can go with the observed raw-scale window. This design leaves fewer intermediate choices and makes the reference experiment easier to reproduce.

Reviewer-directed analyses were performed without changing the main twelve-scenario protocol. Saved test probabilities were used for threshold sensitivity and label-wise evaluation, and a matched five-run baseline was trained for paired statistical tests. These additions strengthen the analysis while preserving the original data-generation and evaluation scope.

V. RESULTS AND DISCUSSION

A. Overall Multi-Label Performance

Table VII reports the main five-run result for the retained configuration. Mean exact match is 96.11%, F1-score is 96.44%, and Hamming loss is 1.04%. The narrow confidence intervals indicate that the result is not driven by one favorable initialization.

Hamming loss gives a label-level view of the error. A value of 1.04% means that only a small fraction of individual label decisions are wrong, even when a full label vector is counted as incorrect by exact match. A separate five-run verification performed for the reviewer analyses produced 96.05% exact match and 96.39% F1-score, which is close to the original result and supports reproducibility.

B. Label-Wise and Pattern-Level Results

Fig. 5 shows precision, recall, and F1-score for each output label using the ensemble mean probability across the five verification runs. Cyclic and systematic labels are the strongest,

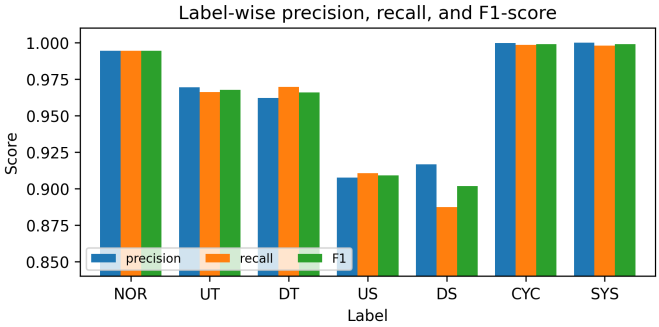

Fig. 5. Label-wise precision, recall, and F1-score.

TABLE VIII. PATTERN-LEVEL EXACT MATCH PERFORMANCE

Scenario	Exact match (%)	Test windows
Downward shift	88.16	1250
Upward trend	90.00	1250
Downward trend	91.04	1250
Upward shift	91.36	1250
Systematic	99.36	1250
Upward trend + cyclic	99.36	1250
Cyclic	99.44	1250
Normal	99.60	1250
Downward trend + cyclic	99.68	1250
Downward trend + systematic	99.68	1250
Upward trend + systematic	99.68	1250
Systematic + cyclic	99.92	1250

TABLE IX. COMPARISON OF EXPLORED CONFIGURATIONS

Configuration	Exact match (%)	F1-score (%)	Hamming loss (%)
Baseline CNN	96.08	96.40	1.04
Two-channel input variant	95.83	96.17	1.11
Targeted-training CNN	96.11	96.44	1.04
Weighted-loss variant	95.95	96.35	1.06
Shift-extra-only variant	95.90	96.24	1.09

with F1-scores of 99.90% and 99.89%, respectively. Upward and downward trend remain above 96.5% F1. Shift labels are weaker: upward shift reaches 90.89% F1 and downward shift 90.16%. The downward-shift false-negative rate is 11.28%, compared with 8.96% for upward shift.

Table VIII and Fig. 6 give the original scenario-level exact match rates. The selected concurrent cases are recognized well, whereas the weaker cases are single shifts and trends. This difference is plausible in a 32-point window. A weak trend may resemble random drift, while a small or late shift leaves only a limited number of post-change observations. In contrast, a cycle or systematic alternation creates repeated evidence across the window.

C. Comparison of Explored Configurations

Table IX summarizes the configurations examined before the retained setup was selected. The differences between the strongest variants are small. The retained targeted-training model is therefore treated as a balanced configuration, not as a universally superior architecture.

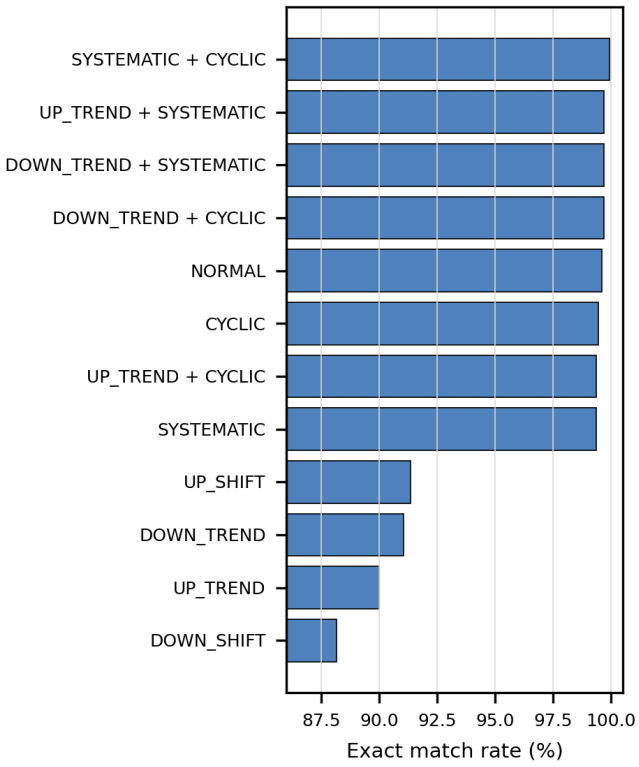

Fig. 6. Pattern-level exact match rate for the twelve control chart scenarios.

To test whether the small baseline difference was meaningful, a new matched five-run comparison was performed using the same seeds. The repeat means were 95.99% versus 96.05% for exact match and 96.33% versus 96.39% for F1-score. Paired t-tests gave $p = 0.558$ for exact match and $p = 0.653$ for F1-score; Wilcoxon tests led to the same conclusion. The difference is not statistically significant. We therefore retain the targeted-training setup because it gave the most balanced behavior across the difficult cases, while avoiding a claim of statistical superiority.

D. Decision-Threshold Sensitivity

Fig. 7 evaluates one common threshold from 0.30 to 0.70. Lower thresholds increase recall and give the highest F1-score at 0.30 (96.96%), but strict exact match falls to 93.98% because more false labels are activated. The threshold of 0.5 gives the highest exact match (96.05%) and the lowest Hamming loss (1.05%) in the repeated analysis. It is therefore a reasonable common operating point for the present objective.

E. Comparison with Prior Concurrent CCPR Studies

The closest published study is the multi-label CNN of Cheng et al. [23]. Table X provides a direct numerical context using their first experiment, which also used 32-point windows, seven labels, 5000 training samples per scenario, and 1250 test samples per scenario. Their protocol contains more concurrent scenarios, whereas the present protocol concentrates on five selected mixtures and preserves raw level information for shifts. The comparison is therefore informative but not a controlled head-to-head benchmark.

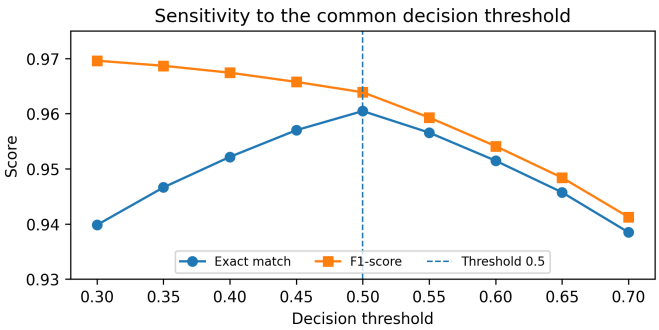

Fig. 7. Sensitivity of exact match and F1-score to the common decision threshold.

TABLE X. CONTEXTUAL COMPARISON WITH THE CLOSEST MULTI-LABEL CNN STUDY

Study	Scenarios	Window	Exact match (%)	F1 (%)	Hamming loss (%)
Cheng et al. [23], Exp. 1	16	32	92.56	95.68	1.98
Present study	12	32	96.11	96.44	1.04

The numerical values should not be read as proof of general superiority because the pattern generators and scenario sets differ. Cheng et al. also included a real-world monitoring demonstration, which is an important strength of their study. The contribution here is narrower: a compact raw-scale reference, explicit repeated-run stability, focused analysis of difficult shift and trend labels, and a transparent evaluation of the selected protocol. Earlier SSA–SVM, ICA–SVM, wavelet, and ELM studies remain relevant, but their composite-class outputs and different generators make direct metric comparison unreliable [15]–[22].

F. Computational Cost

The retained network is lightweight: it has 15,687 trainable parameters, about 54,680 MACs per sample, and a saved model size of approximately 0.22 MB. In the Colab CPU reruns used for the reviewer analysis, mean training time was 582 ± 117 s. Batched inference over the 15,000-window test set took approximately 0.29 s, corresponding to about 0.019 ms per sample. These values are implementation-dependent, but they show that the compact model has modest computational requirements.

G. Limitations and Practical Scope

The main results are obtained from synthetic signals. This controlled setting is useful because the active label vector is known exactly, which is rarely available for concurrent abnormalities in real plant data. However, it does not demonstrate industrial generalization. Sensor effects, autocorrelation, changes in variance, maintenance actions, and sampling changes may alter the observed shapes.

The scenario set is also intentionally limited to five selected concurrent combinations. Shift-containing mixtures, combinations with more than two simultaneous abnormalities, and systematic variation of the noise level were not included in the present protocol. A scientifically consistent extension would require a broader factorial design that controls shift magnitude, change-point position, component amplitudes, temporal alignment, and noise intensity in both training and testing. Adding a small number of post hoc cases would instead evaluate a different problem and would not provide a fair estimate of the model under a properly designed extended protocol. The claims of this study are therefore restricted to the twelve scenarios defined in Section III.

In practical use, the output should be treated as decision support rather than a closed diagnosis. A high probability means that the window resembles a learned component, not that the physical root cause has been proven. Real production validation with expert-confirmed multi-label annotations, broader concurrent combinations, and controlled noise studies remain necessary next steps.

VI. CONCLUSION

This study studied concurrent abnormal pattern recognition in control charts using a direct multi-label CNN. Each window is represented by seven elementary labels, so more than one abnormal behavior can be active without creating a separate composite class for every mixture.

Under the controlled twelve-scenario protocol, the retained model achieved 96.11% exact match accuracy, 96.44% F1-score, and 1.04% Hamming loss across five repeated trainings. Confidence intervals were narrow, and threshold analysis supported 0.5 as a stable common operating point when strict full-vector correctness is prioritized. Label-wise analysis showed very strong cyclic and systematic recognition, while short shifts and trends remained more difficult.

The method should therefore be understood as a compact and interpretable reference for the selected concurrent cases, not as a complete solution for every possible mixture. Shift-containing concurrent patterns, combinations with more than two abnormalities, varying noise conditions, and real industrial data require a dedicated extension of the generation and training protocol. These directions are important future work, but they do not change the main conclusion obtained for the controlled twelve-scenario setting.

DISCLOSURE AND CONFLICT OF INTEREST

We declare no conflict of interest. No financial or personal relationship influenced the design, analysis, or reporting of this work.

DECLARATION ON GENERATIVE AI

We used language and formatting tools only to support manuscript preparation. We reviewed the manuscript and remain responsible for its scientific content.

REFERENCES

[1] D. C. Montgomery, *Introduction to Statistical Quality Control*, 8th ed. Hoboken, NJ, USA: Wiley, 2020.

[2] Western Electric Company, *Statistical Quality Control Handbook*. Indianapolis, IN, USA: Western Electric Company, 1956.

[3] W. A. Shewhart, *Economic Control of Quality of Manufactured Product*. New York, NY, USA: D. Van Nostrand, 1931.

[4] L. S. Nelson, "The Shewhart control chart tests for special causes," *Journal of Quality Technology*, vol. 16, no. 4, pp. 237-239, 1984.

[5] L. S. Nelson, "Interpreting Shewhart control charts," *Journal of Quality Technology*, vol. 17, no. 2, pp. 114-116, 1985.

[6] C. S. Cheng, "A neural network approach for the analysis of control chart patterns," *International Journal of Production Research*, vol. 35, no. 3, pp. 667-697, 1997.

[7] R. S. Guh and J. D. T. Tannock, "Recognition of control chart concurrent patterns using a neural network approach," *International Journal of Production Research*, vol. 37, no. 8, pp. 1743-1765, 1999.

[8] F. Zorriassatine and J. D. T. Tannock, "A review of neural networks for statistical process control," *Journal of Intelligent Manufacturing*, vol. 9, pp. 209-224, 1998.

[9] W. Hachicha and A. Ghorbel, "A survey of control-chart pattern-recognition literature (1991-2010) based on a new conceptual classification scheme," *Computers & Industrial Engineering*, vol. 63, no. 1, pp. 204-222, 2012.

[10] A. Hassan, M. S. N. Baksh, A. M. Shaharoun, and H. Jamaluddin, "Improved SPC chart pattern recognition using statistical features," *International Journal of Production Research*, vol. 41, no. 7, pp. 1587-1603, 2003.

[11] D. T. Pham and M. A. Wani, "Feature-based control chart pattern recognition," *International Journal of Production Research*, vol. 35, no. 7, pp. 1875-1890, 1997.

[12] S. K. Gauri and S. Chakraborty, "Recognition of control chart patterns using improved selection of features," *Computers & Industrial Engineering*, vol. 56, no. 4, pp. 1577-1588, 2009.

[13] V. Ranaee, A. Ebrahimzadeh, and R. Ghaderi, "Application of the PSO-SVM model for recognition of control chart patterns," *ISA Transactions*, vol. 49, no. 4, pp. 577-586, 2010.

[14] Y. Al-Assaf, "Multi-resolution wavelets analysis approach for the recognition of concurrent control chart patterns," *Quality Engineering*, vol. 17, no. 1, pp. 11-21, 2005.

[15] C. J. Lu, Y. E. Shao, and C. C. Li, "Recognition of concurrent control chart patterns by integrating ICA and SVM," *Applied Mathematics & Information Sciences*, vol. 8, no. 2, pp. 681-689, 2014.

[16] L. Xie, N. Gu, D. Li, Z. Cao, M. Tan, and S. Nahavandi, "Concurrent control chart patterns recognition with singular spectrum analysis and support vector machine," *Computers & Industrial Engineering*, vol. 64, no. 1, pp. 280-289, 2013.

[17] S. Du, D. Huang, and J. Lv, "Recognition of concurrent control chart patterns using wavelet transform decomposition and multiclass support vector machines," *Computers & Industrial Engineering*, vol. 66, no. 4, pp. 683-695, 2013.

[18] N. Gu, Z. Cao, L. Xie, D. Creighton, M. Tan, and S. Nahavandi, "Identification of concurrent control chart patterns with singular spectrum analysis and learning vector quantization," *Journal of Intelligent Manufacturing*, vol. 24, pp. 1241-1252, 2013.

[19] W. A. Yang, W. Zhou, W. Liao, and Y. Gou, "Identification and quantification of concurrent control chart patterns using extreme-point symmetric mode decomposition and extreme learning machines," *Neurocomputing*, vol. 147, pp. 260-270, 2015.

[20] G. D. Pelegrina, L. T. Duarte, and C. Jutten, "Blind source separation and feature extraction in concurrent control charts pattern recognition," *Computers & Industrial Engineering*, vol. 92, pp. 105-114, 2016.

[21] C. J. Lu, Y. E. Shao, and P. H. Li, "Mixture control chart patterns recognition using independent component analysis and support vector machine," *Neurocomputing*, vol. 74, no. 11, pp. 1908-1914, 2011.

[22] Z. Zhou, P. Jiang, and X. Wang, "PCA-WA based approach for concurrent control chart pattern recognition," *Computers & Industrial Engineering*, vol. 104, pp. 1-12, 2017.

[23] C. S. Cheng, P. W. Chen, and Y. Ho, "Control chart concurrent pattern classification using multi-label convolutional neural networks," *Applied Sciences*, vol. 12, no. 2, Art. 787, 2022.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[25] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436-444, 2015.

[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[27] F. Chollet, *Deep Learning with Python*. Shelter Island, NY, USA: Manning, 2018.

[28] Z. Wang, W. Yan, and T. Oates, "Time series classification from scratch with deep neural networks: A strong baseline," in *Proc. IJCNN*, 2017, pp. 1578-1585.

[29] J. W. Huang, P. J. Lee, and B. P. Jaysawal, "Multiscale control chart pattern recognition using histogram-based representation of value and zero-crossing rate," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 10, pp. 10751-10761, 2022.

[30] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: https://www.tensorflow.org/

[31] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

[32] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *International Journal of Data Warehousing and Mining*, vol. 3, no. 3, pp. 1-13, 2007.

[33] M. L. Zhang and Z. H. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819-1837, 2014.

[34] F. Herrera, F. Charte, A. J. Rivera, and M. J. del Jesus, *Multilabel Classification: Problem Analysis, Metrics and Techniques*. Cham, Switzerland: Springer, 2016.

[35] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proc. Int. Conf. Engineering and Technology*, 2017, pp. 1-6.

[36] 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," *Computers & Industrial Engineering*, vol. 182, Art. 109410, 2023.

[37] Y. Li, W. Dai, and Y. He, "Control chart pattern recognition under small shifts based on multi-scale weighted ordinal pattern and ensemble classifier," *Computers & Industrial Engineering*, vol. 189, Art. 109940, 2024.

[38] T. Zan, J. Chen, Z. Sun, Z. Liu, M. Wang, X. Gao, and P. Gao, "Pattern recognition of control chart with variable chain length based on recurrent neural network," *Advanced Manufacturing*, vol. 1, Art. 0003, 2024.

[39] T. Zan, X. Jia, X. Guo, M. Wang, X. Gao, et al., "Research on variable-length control chart pattern recognition based on sliding window method and SECNN-BiLSTM," *Scientific Reports*, vol. 15, Art. 5921, 2025.