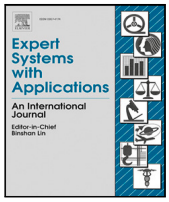
Expert Systems With Applications 249 (2024) 123682

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



An imbalance-aware BiLSTM for control chart patterns early detection

Mohammad Derakhshi, Talayeh Razzaghi *

School of Industrial and Systems Engineering, University of Oklahoma, Norman, OK, USA

ARTICLE INFO

Dataset link: https://github.com/Deraxsi/Imbalance-Aware-LSTM-CCP-Early-Detection.git

Keywords:
Long short-term memory
Control chart pattern recognition
Imbalanced learning
Early detection

ABSTRACT

Digital twins-based predictive models find their roots in smart manufacturing. However, their potential applications to control chart pattern recognition (CCPR) algorithms, which lie at the heart of advanced fault detection systems, remain underexplored. A key challenge in CCPR models arises from the intrinsic imbalance between classes, which can compromise the model's performance if left untreated. Further, existing CCPR models are often trained over simulated control chart data in which abnormal patterns are generated separately from abnormal signals; the resulting classifiers, however, perform poorly in the early detection of abnormalities in real-time production environments. To address these challenges, we develop a cost-sensitive, bi-directional long short-term memory neural network for data sequences with mixed normal and abnormal signals. We further introduce a novel adaptive weighting strategy for data generation by enforcing the rates of abnormal signals within mini-batch distributions. Our model benefits from a new bi-objective early stopping technique, which optimally balances loss minimization and G-mean maximization for model training. Finally, we introduce a novel rolling window-based metric for evaluating CCPR classifier stability. We conduct a comprehensive experimental study of our model using both simulated data and two real-world datasets collected from biomanufacturing and wafer industries. The results of our study consistently demonstrate the superiority of our proposed stopping technique over traditional methods. Our experiments further show the effectiveness of our proposed model in maintaining the classifier stability and specifying optimal process monitoring window length within the datasets.

1. Introduction

A Digital Twin (DT) is a paradigm that establishes a connection between a physical entity, process, human, or human-related feature, and its twin, which is represented as a set of virtual machines or computer-based models. This connection enables monitoring, control, and even optimization of the twin's functions by allowing real-time data exchange between a DT and its physical counterpart. AI-based models can subsequently be utilized for prediction and recognition tasks. As a result, DTs maintain a constant awareness of their surrounding environments, enabling them to stay updated on the events unfolding in real time. Hence, DTs can effectively be utilized in manufacturing systems to continuously monitor their processes, detecting abnormal patterns in order to mitigate the impact of disruptions and prevent expensive failures (Barricelli, Casiraghi, & Fogli, 2019).

The capacity of DTs to perform predictive maintenance enhances the quality control (QC) processes, particularly when equipped with methodologies for control chart pattern recognition (CCPR). In line with this, CCPR algorithms aim to automatically distinguish patterns of out-of-control (abnormal) processes from in-control (normal) ones. The taxonomy of basic out-of-control and in-control patterns was initially proposed by the Western Electric Company (Company, 1956; Hwarng & Hubele, 1993b). It encompasses the following fundamental patterns: (1) normal, (2) up-trend, (3) down-trend, (4) up-shift, (5) down-shift, (6) cyclic, (7) systematic, and (8) stratification patterns; see Fig. 1. Each of these abnormal patterns represents a distinct malfunction in a manufacturing system (Xanthopoulos & Razzaghi, 2014). Thus, a well-designed DT, equipped with CCPR, has the capability to predict these abnormalities accurately and subsequently execute appropriate actions to minimize maintenance costs.

Over the years, researchers have proposed various approaches to address the CCPR problem. Hachicha and Ghorbel (2012) provide a comprehensive review of several approaches developed between 1991 and 2010 for CCPR, which are categorized into two main groups: rule-based expert systems and artificial neural network (ANN)-based methods, including both supervised and unsupervised techniques within the ANN framework. Rule-based expert systems methods diagnose abnormality patterns by analyzing the statistical features of raw data (Kuo & Mital, 1993; Pham & Oztemel, 1992; Swift & Mize, 1995). Because

* Corresponding author.
E-mail addresses: deraxsi@ou.edu (M. Derakhshi), talayeh.razzaghi@ou.edu (T. Razzaghi).

https://doi.org/10.1016/j.eswa.2024.123682
Received 10 November 2023; Received in revised form 29 February 2024; Accepted 13 March 2024
Available online 26 March 2024

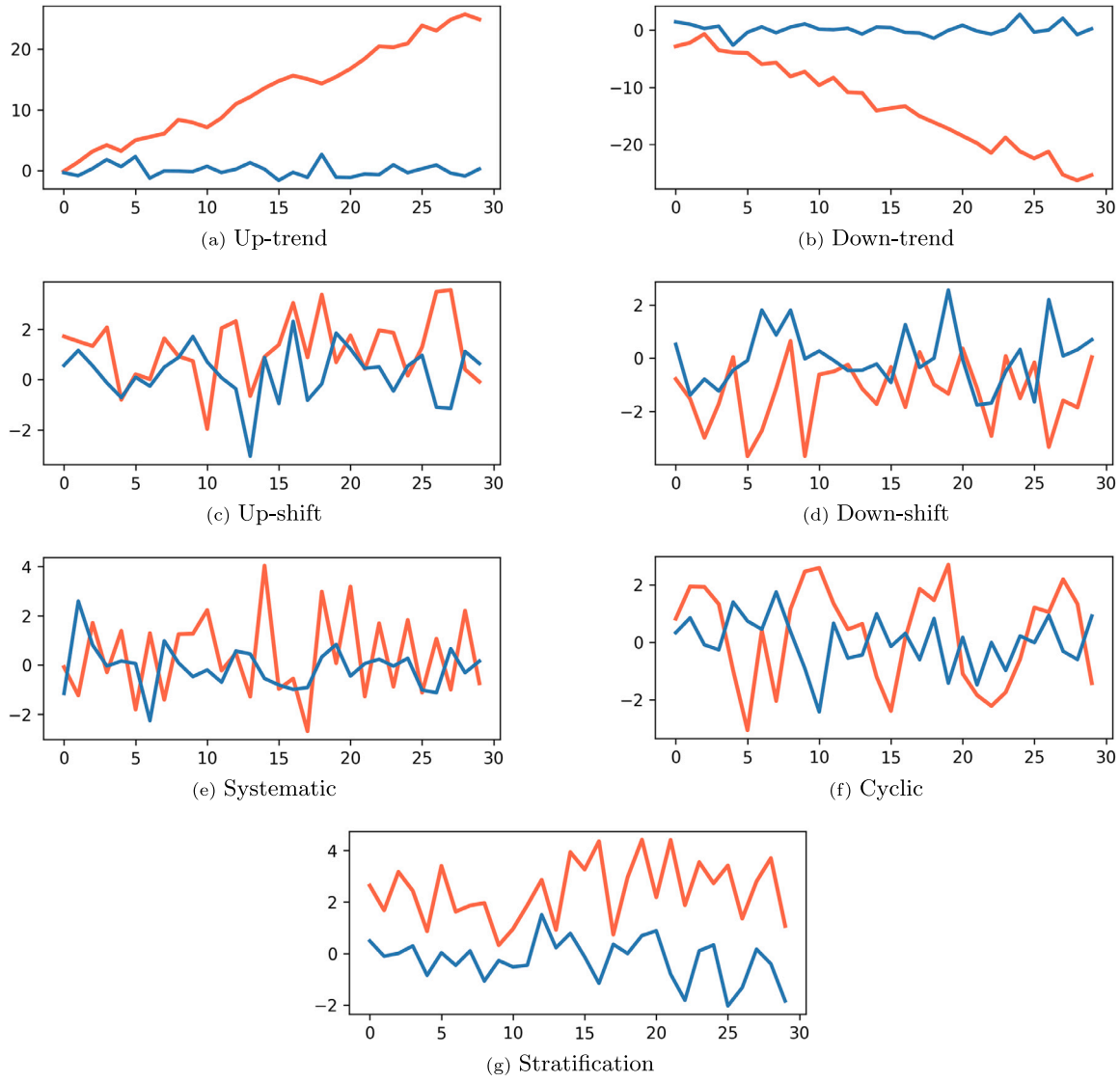


Fig. 1. Illustration of the basic abnormal patterns (red) against a normal one (blue).

such methods rely on limited statistical features, they suffer from frequent false recognition when the extracted statistical features remain similar despite originating from different patterns. ANN methods formulate CCPR as an imbalanced learning problem, accounting for unevenly distributed data points from various patterns. Recently, Tran, Ahmadi Nadi, Nguyen, Tran, and Tran (2022) studied three main categories of AI-based algorithms: regression and decision trees, ANNs and deep learning (DL), and support vector machines (SVM).

Several DL-based works have been published within the CCPR literature. Recently, Fuqua and Razzaghi (2020) studied the binary-class CCPR problem using convolutional neural networks (CNNs). Miao and Yang (2019) investigated multi-class CCPR by considering fourteen classes including basic and composite patterns. Zan, Liu, Wang, Wang, and Gao (2020) further explored a similar problem by specifically focusing only on six basic patterns. Yu, Zheng, and Wang (2019) studied a deep denoising autoencoder to extract decisive features for the classification task. Shao and Chiu (2016) conducted a study to compare different methods, including deep ANN- and random forest-based models, for detecting mixed abnormal patterns. Lu, Wang, and Dai (2020) introduced an information fusion-assisted framework for CCPR with dynamically-sized window lengths using data from multiple sensors. The framework involves concatenating the outputs of CCPR models for multiple signals into a vector, which serves as the input for the classification model. In a similar work, Maged and Xie (2023) developed a CNN-based model to tackle a CCPR problem with varying window lengths of input data. To convert variable-sized inputs into fixed-sized input vectors, they employ a resampling-based signal resizing followed by a CNN.

Further, recurrent neural networks (RNNs) (Hopfield, 1982; Jordan, 1997) and long short-term memory networks (LSTM) (Hochreiter & Schmidhuber, 1997) have been introduced for CCPR problems. Pacella and Semeraro (2007) conducted a study on a variant of RNN inspired by Elman (1990) to detect abnormal shifts. Liu, Chen, Jin, and Qu (2019) studied an LSTM-based network for identifying abnormalities in the body-in-white manufacturing process. Additionally, a well-known variant of LSTM known as bi-directional LSTM (BiLSTM), has been developed for CCPR problems. For example, Ünlü (2021a) employed a cost-sensitive BiLSTM using the conventional synthetic data generation framework and then compared it with CNN and SVM models. The BiLSTM was originally introduced by Graves and Schmidhuber (2005) which employs forward and backward LSTMs to capture both past and future context information. This structure allows the model to leverage additional information when processing earlier time steps in reverse order. Furthermore, BiLSTM has demonstrated remarkable success in effectively handling sequence data across various fields, including time series classification (Bao, Yue, & Rao, 2017; Graves & Schmidhuber,

2005), natural language processing (Chen, Xu, He, & Wang, 2017), speech recognition (Graves, Mohamed, & Hinton, 2013), and handwriting recognition (Kizilirmak & Yanikoglu, 2023). Indeed, studies suggest that BiLSTM often outperforms LSTM (Graves et al., 2008), particularly in the context of time series classification problems. Hence, this study utilizes BiLSTM. In particular, we employ a BiLSTM equipped with peephole connections due to its improved information flow, which enhances the model's ability to capture long-term dependencies and consequently improves overall performance (Wei, Xu, & Hu, 2021; Xu, Du, & Zhang, 2019).

The imbalanced nature of data presents a significant challenge in effectively training DL models for classification problems (Wang, Minku, & Yao, 2018; Wang, Zhao, Jiang, Gao, & BNRist, 2018; Zhang et al., 2020) in which the highly interested classes (abnormal class), have significantly fewer samples compared to the other class (normal class). When developing DL-based models for imbalanced class data, the distribution of data may change dynamically within each mini-batch (Li, Liu, & Wang, 2019; Lu, Cheung, & Tang, 2017; Malialis, Panayiotou, & Polycarpou, 2020). However, the abnormal data pattern is usually assumed to remain fixed during training. Consequently, the deep model, with its fixed class imbalanced and abnormal pattern, lacks the ability to adapt to the dynamic shifts in data distribution. To the best of our knowledge, there is no prior work that addresses this issue in CCPR when learning LSTM models.

While minimizing loss and maximizing scores are closely related concepts, it is suggested in general to avoid directly maximizing the score during network training. This is because the score often represents a discontinuous function in relation to the model's predictions. Consequently, alternative strategies have been introduced to address this issue in the literature (Marchetti, Guastavino, Piana, & Campi, 2022). In this context, we present a new bi-objective early stopping technique, aiming to simultaneously achieve lower loss and higher G-mean values. This technique updates the current solution when a weighted combination of improvement in validation loss and G-mean values exceeds the current solution. Otherwise, it terminates if the number of epochs without improvement surpasses the specified threshold.

Further, conventional methods (Ünlü, 2021a; Xue, Wu, Zheng, & He, 2023) for simulating control chart data typically involve generating abnormal patterns that solely consist of abnormal signals. However, a classifier that is trained solely on data samples with abnormal signals may encounter challenges in the early detection of abnormalities in a real-time production line. This challenge arises from the fact that the overwhelming presence of normal signals can overshadow the existence of abnormal signals.

Our contributions are summarized as follows:

- We develop a new cost-sensitive CCPR technique based on BiLSTM that addresses imbalanced-class data. Our approach acknowledges the inherent differences within abnormal pattern data. Our model accounts for the initiation time of abnormal patterns in each sample as well as the imbalanced distribution of classes within the mini-batches.
- We develop a novel bi-objective early stopping technique designed to optimally balance both loss minimization and G-mean maximization simultaneously during the training of BiLSTM networks. This innovative approach addresses the challenges associated with score maximization due to its discontinuous nature.
- We introduce a novel model selection approach that considers multiple realizations of abnormality instances, encompassing varying rates, and window lengths. Within this framework, our approach allows the BiLSTM models to adapt to the specific characteristics of each abnormality pattern.
- We introduce an innovative rolling-window-based metric based on a data generation technique (Ünlü, 2021b) to assess the stability of a CCPR classifier. This metric leverages the maximum streak of true alerts from the initial detection points and enables the determination of the optimal window length for achieving reliable early detection.
- Finally, we implement and validate our DT-assisted CCPR model for real-world datasets collected from (1) a wafer manufacturing industry, and (2) a bio-manufacturing industry.

Table 1
Mathematical models of control chart patterns.

Pattern	Mathematical Models	
	AUDS	MRDS
Normal	$x_t = \mu + \epsilon_t$	$x_t = \mu + \epsilon_t$
Up-trend	$x_t = \mu + \epsilon_t + \xi_1 t$	$x_t = \begin{cases} \mu + \epsilon_t, & \text{if } t < t_0 \\ \mu + \epsilon_t + \xi_1 (t - t_0), & \text{if } t \geq t_0 \end{cases}$
Down-trend	$x_t = \mu + \epsilon_t - \xi_1 t$	$x_t = \begin{cases} \mu + \epsilon_t, & \text{if } t < t_0 \\ \mu + \epsilon_t - \xi_1 (t - t_0), & \text{if } t \geq t_0 \end{cases}$
Up-shift	$x_t = \mu + \epsilon_t + \xi_2$	$x_t = \begin{cases} \mu + \epsilon_t, & \text{if } t < t_0 \\ \mu + \epsilon_t + \xi_2, & \text{if } t \geq t_0 \end{cases}$
Down-shift	$x_t = \mu + \epsilon_t - \xi_2$	$x_t = \begin{cases} \mu + \epsilon_t, & \text{if } t < t_0 \\ \mu + \epsilon_t - \xi_2, & \text{if } t \geq t_0 \end{cases}$
Systematic	$x_t = \mu + \epsilon_t - \xi_3 (-1)^t$	$x_t = \begin{cases} \mu + \epsilon_t, & \text{if } t < t_0 \\ \mu + \epsilon_t + \xi_3 (-1)^t, & \text{if } t \geq t_0 \end{cases}$
Cyclic	$x_t = \mu + \epsilon_t + \xi_4 \sin(\frac{2\pi t}{\Omega})$	$x_t = \begin{cases} \mu + \epsilon_t, & \text{if } t < t_0 \\ \mu + \epsilon_t + \xi_4 \sin(\frac{2\pi (t - t_0)}{\Omega}), & \text{if } t \geq t_0 \end{cases}$
Stratification	$x_t = \mu + \epsilon_t + \xi_5 \epsilon'_t$	$x_t = \begin{cases} \mu + \epsilon_t, & \text{if } t < t_0 \\ \mu + \epsilon_t + \xi_5 \epsilon'_t, & \text{if } t \geq t_0 \end{cases}$

This paper is organized as follows. Section 2 describes the synthetic data generation method, the proposed cost-sensitive BiLSTM with a new weighting strategy, and the proposed model selection approach. Section 3 discusses the computational results. The conclusion and insights into future research avenues are presented in Section 4.

2. Methodology

2.1. Synthetic data generation

Traditional data generation methods in CCPR often create sequences that exhibit either completely normal or abnormal signals, as illustrated in Fig. 1. Recent works refer to these conventional methods as usual data simulation (AUDS) in the literature (Ünlü, 2021b). In general, these signal patterns are formulated as $x_t = \mu + \epsilon_t + a(t)$, where x_t is the control chart signal at time t, μ is a constant term (typically assumed to be zero for simplicity), $\epsilon_t \sim \mathcal{N}(0, 1)$ represents a standard normal random term, and $a(t)$ is the abnormal signal at time t which is the function of abnormal patterns (De la Torre Gutiérrez & Pham, 2018; Ünlü, 2021a). The detailed formulations for these patterns, including the normal pattern, are presented in the second column of Table 1. Assume $\xi_j > 0$ for $j = 1, \ldots, 5$, where each parameter uniquely characterizes a specific process abnormalities. Particularly, ξ_1 denotes the slope of trends and reflects the rate at which they either increase or decrease. ξ_2 indicates the magnitude of shifts in shift patterns. ξ_3 assesses the granularity of systematic abnormalities. ξ_4 and Ω represent the amplitude and period of cyclic pattern, respectively, wherein Ω is set to 8 akin to previous studies (Xanthopoulos & Razzaghi, 2014). Finally, ξ_5 describes the intensity of stratification patterns. While the categorization and formulation of these patterns simplifies prediction tasks, it may obscure the predictive algorithm's efficacy given the complexity of real-world data that often exhibits a mixture of these signals.

Our methodology addresses the limitations of the AUDS approach by building upon the method introduced by Wu, Liu, and Zhu (2015), which utilized variable initiation points for abnormalities within shift patterns. We enhance this concept by incorporating the mix ratio data simulation (MRDS) technique from Ünlü (2021b), thereby introducing a

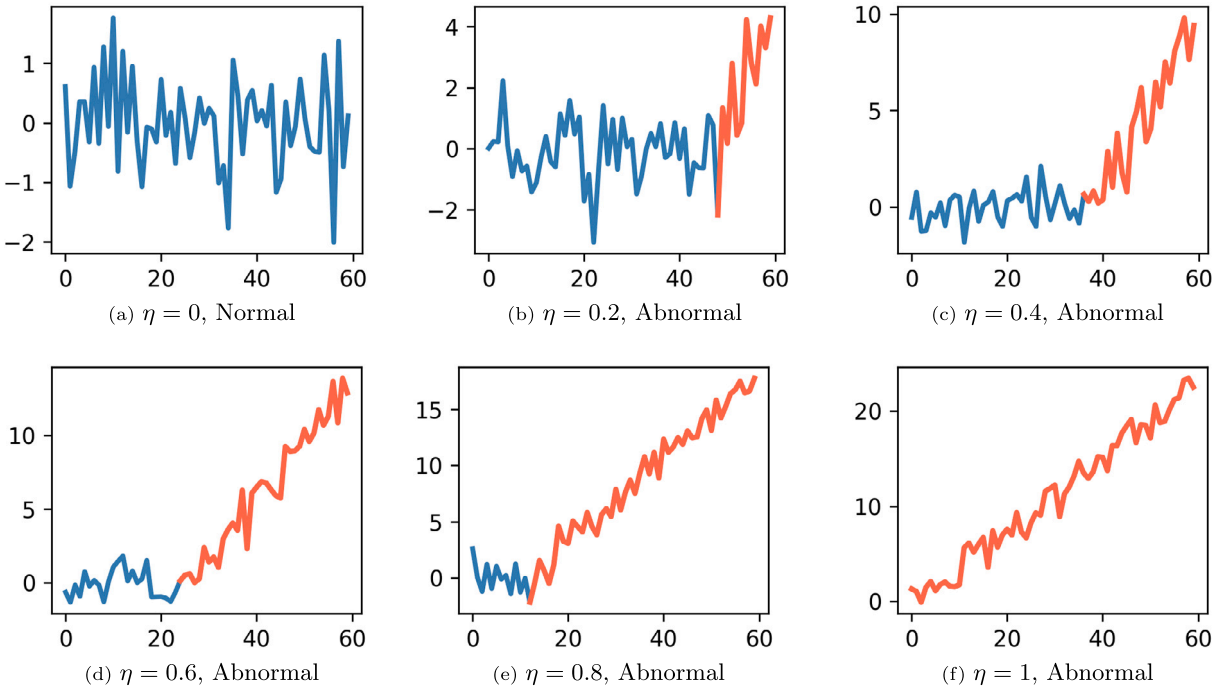


Fig. 2. Example of MRDS synthetic data generation. Within the sequence, the up-trend abnormal signal is highlighted in red, while the normal signal is indicated in blue. Below each data sample, the label—'normal' or 'abnormal'—is clearly annotated, corresponding to the respective η value.

broader spectrum of abnormal signal representations into our datasets. The proportion of abnormalities within a data sequence is controlled by the abnormal signal rate denoted by η. The abnormal signals emerge in the last $\eta\times100\%$ of a sequence of length T, starting from $t_0 = \lfloor T\times(1-\eta)\rfloor$. In this case, the pattern transitions from normal signal ($x_t = \mu + \epsilon_t$) to abnormal signal ($x_t = \mu + \epsilon_t + a(t)$) at t_0, as detailed in the third column of Table 1. For example, the abnormal data point with $\eta = 0.2$ shows the last 20 percent of signals are abnormal. Fig. 2 illustrates six data samples generated using the MRDS approach for various values of η. In particular, Fig. 2(a) presents a normal data sample, and Figs. 2(b)–2(f) show various data samples that are generated by varying η. In this paper, we intend to employ MRDS with $\eta \in \{0.2, 0.4, 0.6, 0.8, 1\}$. This visual representation allows for an immediate identification of the proportion and distribution of abnormal signals embedded in the sequence.

2.2. Bi-directional LSTM

The standard forward LSTM unit with peephole connections consists of a cell, an input gate, an output gate, and a forget gate, which are illustrated in Fig. 3(a). BiLSTM employs forward and backward LSTM that captures both past and future information, subsequently combining these components to generate the final output. In the first (forward) layer, the operations follow the same direction as the data sequence, while in the second (backward) layer, they proceed in the opposite direction.

The input data for CCPR problems is in the form of time series. Assume a data sample $\mathbf{x} = (x_1, x_2, \dots, x_T) \in \mathbb{R}^T$ where T is a positive integer value. The parameter T is commonly referred to as the window length in the context of time series data. The input of a BiLSTM at time t is denoted as x_t such that $x_t = \overrightarrow{h_t^0} = \overleftarrow{h_t^0}$. The $\overrightarrow{h_t^l}$ and $\overleftarrow{h_t^l}$ are forward and backward output components at layer l respectively, given a positive integer number L denoting the number of hidden layers. For $l = 0$, $\overrightarrow{h_t^0}$ and $\overleftarrow{h_t^0}$ are forward and backward components at layer 0 respectively. The BiLSTM can be formulated as follows (Graves & Schmidhuber, 2005; Yu, Si, Hu & Zhang, 2019),

Forward LSTM:

$$\forall l = 1, 2, \dots, L \;\&\; t = 1, 2, \dots, T,$$

$$\overrightarrow{f_t^l} = \sigma(\overrightarrow{W_{fh}^l}\overrightarrow{h_{t-1}^l} + \overrightarrow{W_{fx}^l}\overrightarrow{h_t^{l-1}} + \overrightarrow{P_f^l} \odot \overrightarrow{c_{t-1}^l} + \overrightarrow{b_f^l}), \tag{1}$$

$$\overrightarrow{i_t^l} = \sigma(\overrightarrow{W_{ih}^l}\overrightarrow{h_{t-1}^l} + \overrightarrow{W_{ix}^l}\overrightarrow{h_t^{l-1}} + \overrightarrow{P_i^l} \odot \overrightarrow{c_{t-1}^l} + \overrightarrow{b_i^l}), \tag{2}$$

$$\overrightarrow{\tilde{c}_t^l} = \tanh(\overrightarrow{W_{\tilde{c}h}^l}\overrightarrow{h_{t-1}^l} + \overrightarrow{W_{\tilde{c}x}^l}\overrightarrow{h_t^{l-1}} + \overrightarrow{b_{\tilde{c}}^l}), \tag{3}$$

$$\overrightarrow{c_t^l} = \overrightarrow{f_t^l} \odot \overrightarrow{c_{t-1}^l} + \overrightarrow{i_t^l} \odot \overrightarrow{\tilde{c}_t^l}, \tag{4}$$

$$\overrightarrow{o_t^l} = \sigma(\overrightarrow{W_{oh}^l}\overrightarrow{h_{t-1}^l} + \overrightarrow{W_{ox}^l}\overrightarrow{h_t^{l-1}} + \overrightarrow{P_o^l} \odot \overrightarrow{c_t^l} + \overrightarrow{b_o^l}), \tag{5}$$

$$\overrightarrow{h_t^l} = \overrightarrow{o_t^l} \odot \tanh(\overrightarrow{c_t^l}) \tag{6}$$

Backward LSTM:

$$\forall l = 1, 2, \dots, L \;\&\; t = T, T-1, \dots, 1,$$

$$\overleftarrow{f_t^l} = \sigma(\overleftarrow{W_{fh}^l}\overleftarrow{h_{t+1}^l} + \overleftarrow{W_{fx}^l}\overleftarrow{h_t^{l-1}} + \overleftarrow{P_f^l} \odot \overleftarrow{c_{t+1}^l} + \overleftarrow{b_f^l}), \tag{7}$$

$$\overleftarrow{i_t^l} = \sigma(\overleftarrow{W_{ih}^l}\overleftarrow{h_{t+1}^l} + \overleftarrow{W_{ix}^l}\overleftarrow{h_t^{l-1}} + \overleftarrow{P_i^l} \odot \overleftarrow{c_{t+1}^l} + \overleftarrow{b_i^l}), \tag{8}$$

$$\overleftarrow{\tilde{c}_t^l} = \tanh(\overleftarrow{W_{\tilde{c}h}^l}\overleftarrow{h_{t+1}^l} + \overleftarrow{W_{\tilde{c}x}^l}\overleftarrow{h_t^{l-1}} + \overleftarrow{b_{\tilde{c}}^l}), \tag{9}$$

$$\overleftarrow{c_t^l} = \overleftarrow{f_t^l} \odot \overleftarrow{c_{t+1}^l} + \overleftarrow{i_t^l} \odot \overleftarrow{\tilde{c}_t^l}, \tag{10}$$

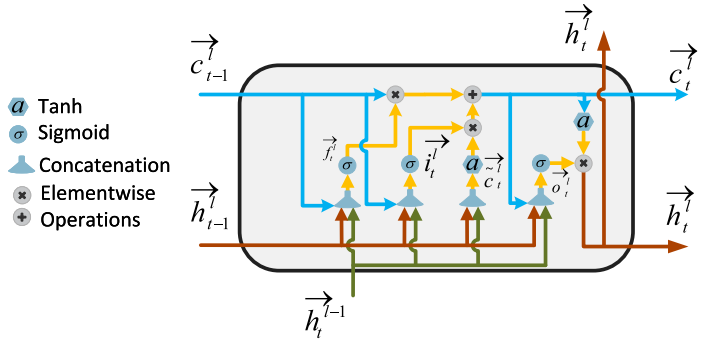
$$\overleftarrow{o_t^l} = \sigma(\overleftarrow{W_{oh}^l}\overleftarrow{h_{t+1}^l} + \overleftarrow{W_{ox}^l}\overleftarrow{h_t^{l-1}} + \overleftarrow{P_o^l} \odot \overleftarrow{c_t^l} + \overleftarrow{b_o^l}), \tag{11}$$

$$\overleftarrow{h_t^l} = \overleftarrow{o_t^l} \odot \tanh(\overleftarrow{c_t^l}) \tag{12}$$

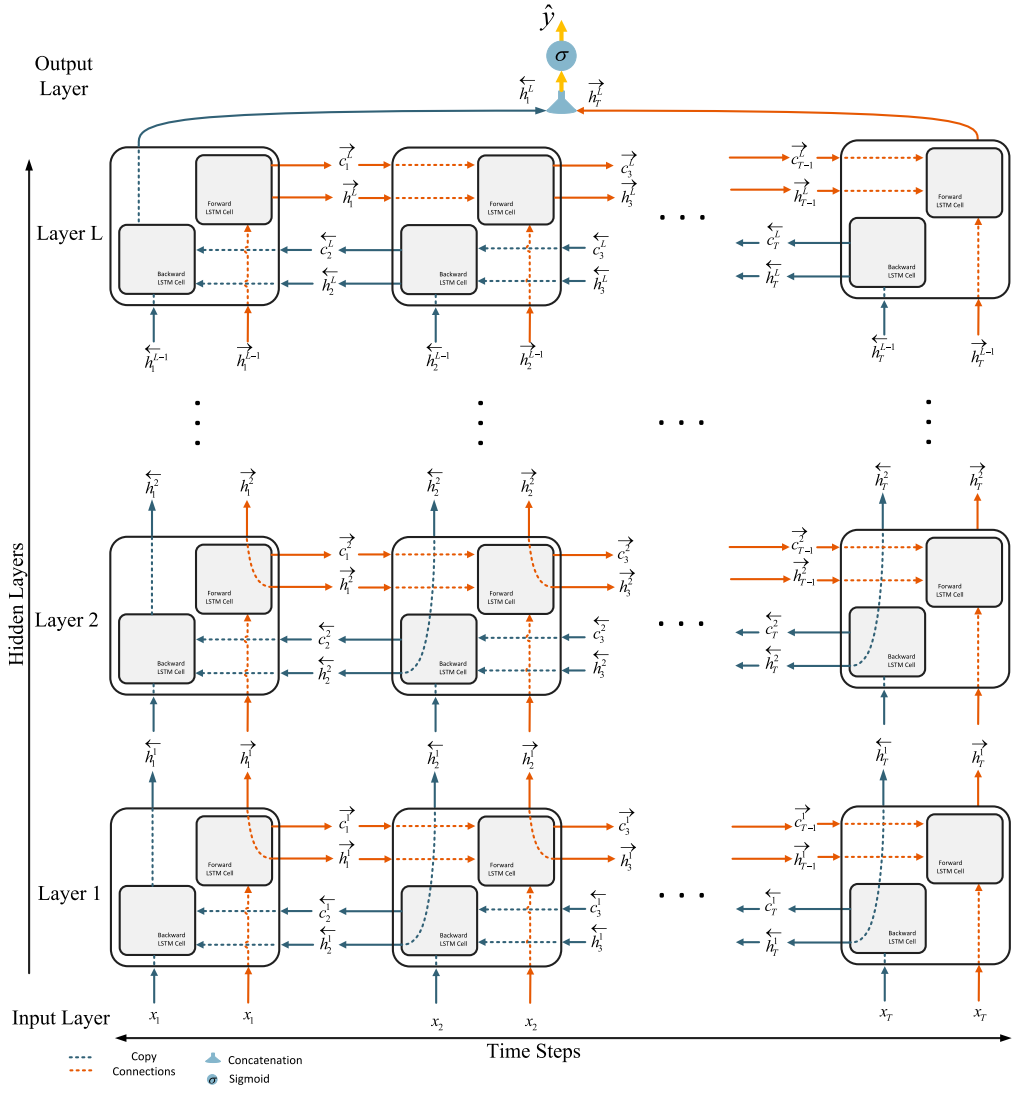
Output Layer L:

$$\hat{y} = \sigma(\overrightarrow{U}\overrightarrow{h_T^L} + \overleftarrow{U}\overleftarrow{h_1^L}) \tag{13}$$

where in the Eq. (1), $\overrightarrow{f_t^l}$ is the forward forget gate at time t and hidden layer l, $\overrightarrow{W_{fh}^l}$ and $\overrightarrow{W_{fx}^l}$ are weight matrices of forward forget gate at hidden layer l, $\overrightarrow{P_f^l}$ is the forward forget peephole weights at hidden layer l, $\overrightarrow{b_f^l}$ is the forward bias term for the forget gate at hidden layer l, σ is the sigmoid activation function, and $\odot$ denotes the element-wise multiplication. In Eq. (2), $\overrightarrow{i_t^l}$ is the forward input gate at time t and hidden layer l, $\overrightarrow{W_{ih}^l}$ and $\overrightarrow{W_{ix}^l}$ are the weight matrices of forward input gate at hidden layer l, $\overrightarrow{P_i^l}$ is the forward input peephole weights at



(a) Forward LSTM cell



(b) Schematic view of our BiLSTM model

Fig. 3. BiLSTM model architecture overview. Fig. 3(a) represents the forward LSTM cell. Fig. 3(b) shows the BiLSTM with a peephole connections.

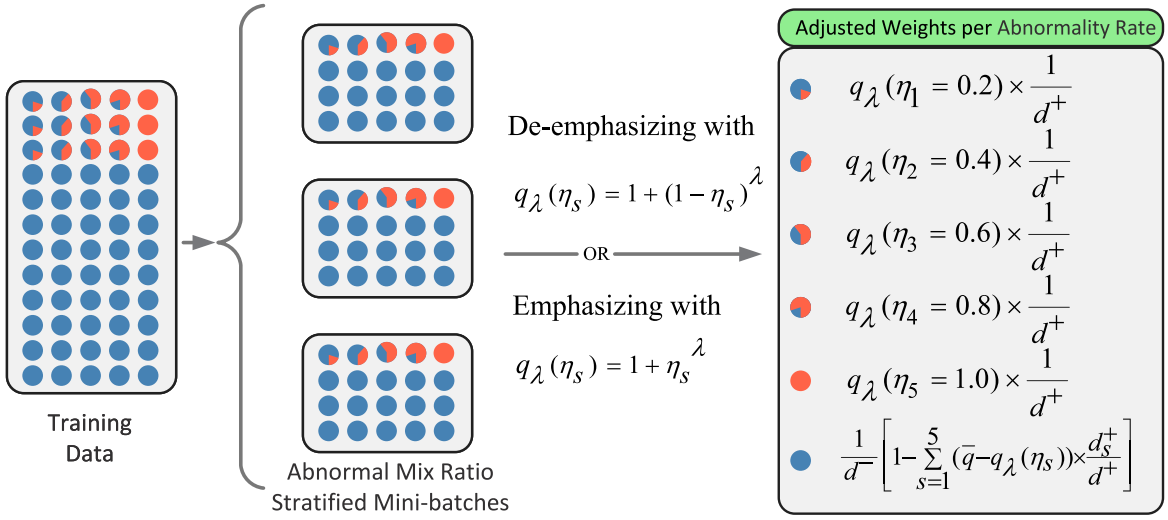


Fig. 4. Schematic representation of abnormal mix ratio stratified mini-batch sampling. The right figure shows the mix ratio weighting approach.

hidden layer l, $\overrightarrow{b_f^l}$ is the forward bias term for the input gate at hidden layer l. In Eq. (3), $\overrightarrow{\tilde{c}_t^l}$ is the forward candidate cell state at time t and hidden layer l, $\overrightarrow{W_{\tilde{c}h}^l}$ and $\overrightarrow{W_{\tilde{c}x}^l}$ are forward weight matrices for candidate cell state at hidden layer l, and $\overrightarrow{b_{\tilde{c}}^l}$ is the forward bias term for candidate cell state at the hidden layer l. $\overrightarrow{c_t^l}$ in Eq. (3) denotes the cell state at time t and hidden layer l. In Eq. (5), $\overrightarrow{o_t^l}$ is the forward output gate at time t and the hidden layer l, $\overrightarrow{W_{oh}^l}$ and $\overrightarrow{W_{ox}^l}$ are the forward weight matrices for output gate at time t and hidden layer l, $\overrightarrow{P_o^l}$ is the forward peephole weights for the output gate, and $\overrightarrow{b_o^l}$ denotes the bias term output gate at the hidden layer l. Finally, Eq. (6) defines $\overrightarrow{h_t^l}$ the forward hidden state at time t and hidden layer l. Similarly, the weights and components of the backward LSTM layers are defined through Eqs. (7)–(12). In Eq. (13), the vectors $\overrightarrow{h_T^L}$ and $\overleftarrow{h_1^L}$ obtained from the forward and backward LSTMs in the final layer L are combined using weights $\overrightarrow{U}$ and $\overleftarrow{U}$. Subsequently, they are activated by the sigmoid function (σ) to calculate the probability ($\hat{p}$) indicating whether the given sequence $\mathbf{x}$ is abnormal.

2.3. BiLSTM learning for CCPR

In this section, we introduce the main components of BiLSTM learning for CCPR and present our novel approaches for each component. First, we formalize our cost-sensitive BiLSTM learning problem (Section 2.3.1). Next, we introduce a sample weighting approach seeking to adjust sample weights based on the mini-batch distributions. Additionally, we introduce a bi-objective early-stopping method that maximizes validation G-mean while it minimizes the validation loss Section 2.5. Lastly, we present our innovative model selection approach in Section 2.6.

2.3.1. Cost-sensitive loss function

Cost-sensitive learning is a well-known approach for learning from imbalanced datasets (Elkan, 2001). The underlying concept in cost-sensitive learning is to assign weights to samples, which results in non-uniform contributions of samples to the trained classifier. Consider a data set $D = \{(\mathbf{x}^{(j)}, y^{(j)}) : j = 1, \ldots, n\}$ with $\mathbf{x}^{(j)} \in \mathbb{R}^T$ and $y^{(j)} \in \{0, 1\}$ being the jth sample and its corresponding ground-truth class label, respectively. A generic cost-sensitive learning problem is represented as

$$\min_{\beta} \quad \sum_{j=1}^{n} \alpha^{(j)} \ell\left(y^{(j)}, \hat{y}^{(j)}\right), \tag{14}$$

where ℓ is the *loss function*, $\hat{y}^{(j)}$ is the classifier's predicted label for sample $j = 1, \ldots, n$, and $\alpha^{(j)}$ represents the weight of the jth sample.

Several loss functions exist in the literature, e.g., the mean-squared error, the hinge loss, and the cross-entropy loss; see Wang, Ma, Zhao, and Tian (2020) for a recent survey of common loss functions in machine learning (Wang et al., 2020). More recently, the focal loss function has been introduced. The focal loss extends the cross-entropy loss and prioritizes training on hard samples, reducing the impact of well-classified and easy samples (Lin, Goyal, Girshick, He, & Dollár, 2017). Accordingly, it has successfully been applied to imbalanced learning in several applications, including image segmentation (Yeung, Sala, Schönlieb, & Rundo, 2022), speech recognition (Zhu, Dai, Hu, & Li, 2020), and time-series analysis (Aljubran, Ramasamy, Albassam, & Magana-Mora, 2021). Hence, in this work, we choose the focal loss function when solving (14). Its mathematical formulation is denoted as

$$\ell\left(y^{(j)}, p^{(j)}\right) = \begin{cases} \left(1 - p^{(j)}\right)^\gamma \log(p^{(j)}), & \text{if } y^{(j)} = 1, \\ \left(p^{(j)}\right)^\gamma \log(1 - p^{(j)}), & \text{otherwise,} \end{cases} \tag{15}$$

where $p^{(j)} = \sigma(\overrightarrow{U}\overrightarrow{h_T^L} + \overleftarrow{U}\overleftarrow{h_1^L})$, $j = 1, \ldots, n$, and $\gamma \geq 0$ is the modulating factor (which is to be tuned as a hyperparameter). Note that $p^{(j)}$ is the output of BiLSTM for the data sample $\mathbf{x}^{(j)}$ (see Eqs. (1)–(13)).

Remark 1. The original focal loss function proposed by Lin et al. (2017) includes a parameter α within the definition of loss function as the primary motivation of the focal loss function is to address the class imbalance. To represent the generic form of imbalanced learning, our definition of the focal loss function excludes α and considers them as cost-sensitive hyperparameters.

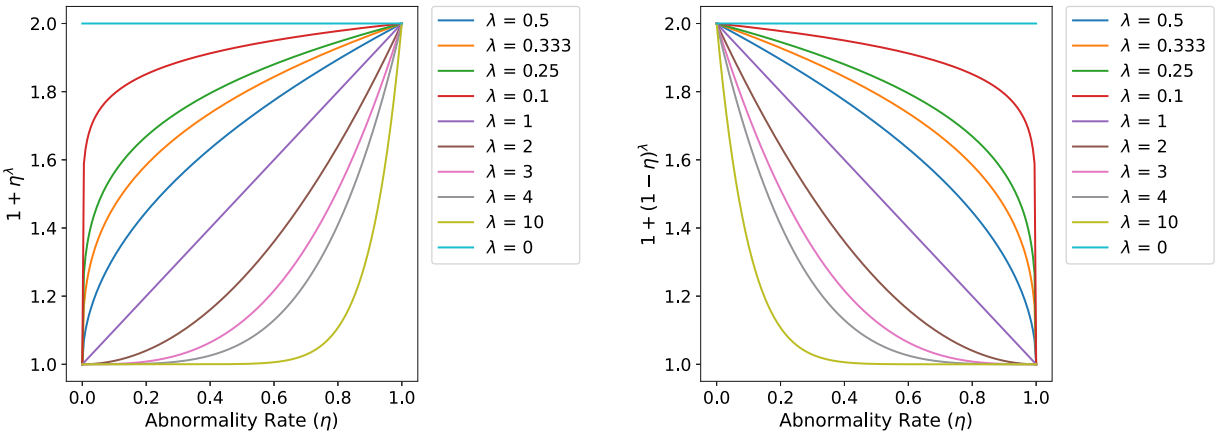
We let the parameter α as

$$\alpha^{(j)} = \begin{cases} \frac{1}{n^+} & \text{if } y^{(j)} = 1 \\ \frac{1}{n^-} & \text{otherwise} \end{cases}$$

where n^+ and n^- are the numbers of abnormal and normal data samples, respectively. We address imbalanced classification by employing a cost-sensitive focal loss function, and we refer to the BiLSTM network using this loss function as the cost-sensitive BiLSTM (CSBiLSTM) throughout this paper.

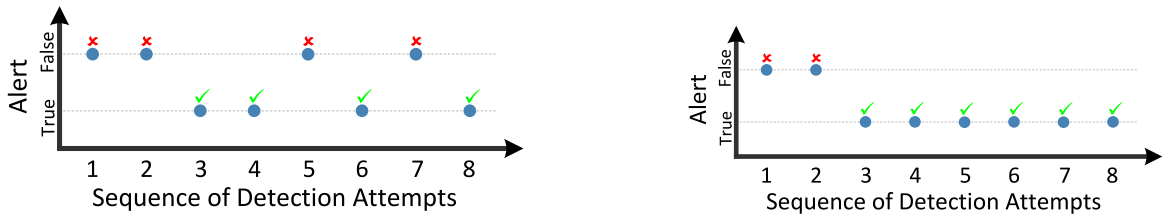
2.3.2. Mix ratio weighting approach

The gradient descent optimization algorithm, particularly the mini-batch gradient descent method, is extensively used in training deep



(a) Emphasizing using $1+\eta^{\lambda}$ (b) De-emphasizing using $1+(1-\eta)^{\lambda}$

Fig. 5. Two leveraged categories of functions for weighting of the abnormal samples.



(a) Less stable classifier (b) More stable classifier

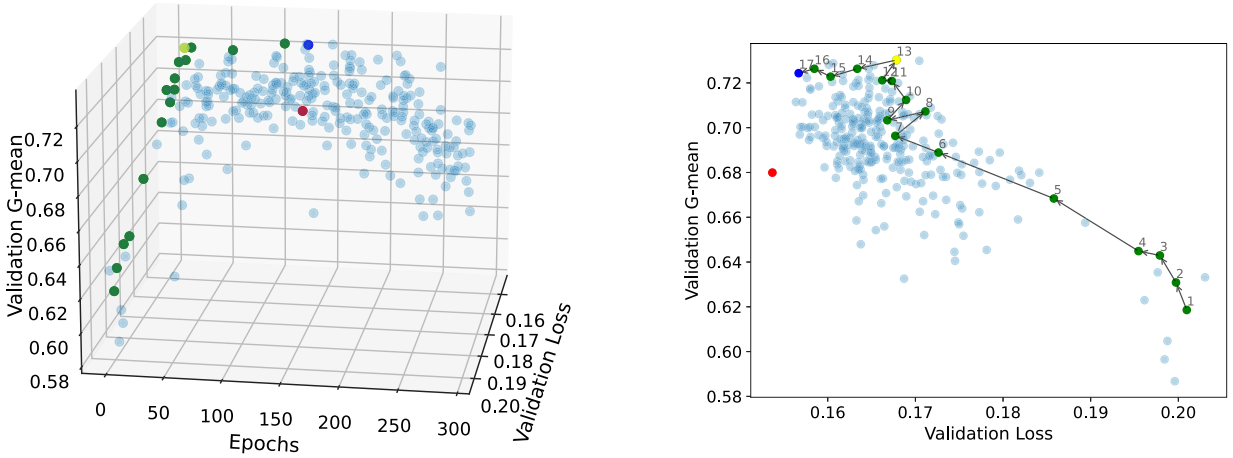
Fig. 6. A comparison between (a) the less stable classifier and (b) the more stable classifier when performing classification on a rolled data point with the rolling window of 8.

learning architectures, including LSTMs (Yu, Si, Hu & Zhang, 2019). However, the mini-batch gradient descent algorithm has limitations when employed in the training of RNNs and RNNs (Dokuz & Tufekci, 2021). A major drawback of the mini-batch gradient descent algorithm is the random selection of data samples, so the distribution of the mini-batches (e.g., imbalance ratio) may differ from that of the original training data. This discrepancy can reduce the performance of the LSTM model, particularly when dealing with imbalanced datasets. To address this issue, previous works have suggested various strategies. The oversampling methods, such as the synthetic minority over-sampling technique (SMOTE) (Chawla, Bowyer, Hall, & Kegelmeyer, 2002; Fernández, Garcia, Herrera, & Chawla, 2018), have been used to create balanced mini-batches by augmenting the size of the minority class. Further, Shimizu et al. (2018) have proposed a balanced mini-batch generating, which randomly creates balanced mini-batches through sampling with replacement. Furthermore, mini-batch gradient descent with stratified sampling has been introduced to address the challenge of imbalanced data (Peng, Gu, Hu, & Liu, 2021). Stratified sampling, a technique commonly employed in statistics, ensures that a specific number of samples are taken from each class or subpopulation. Inspired by this idea and mix ratio data simulation, we have developed the abnormal mix ratio mini-batch gradient descent with stratified sampling. This approach not only preserves the imbalance ratio within each mini-batch during training but also ensures that each mini-batch contains a similar number of abnormal samples with a specified abnormality rate. Fig. 4 demonstrates our approach.

In addition, we aim to further alleviate the class imbalance issue with a new approach known as mix ratio weighting (MRW), tailored to each data sequence with a fixed window length. It is worth noting that the standard weighting strategy treats all abnormal data samples equally by assigning them similar weights. Previous studies have demonstrated that data samples with smaller abnormality values within a sequence are less likely to be detected early by a classifier (Ünlü, 2021a). For instance, when using a window length of 100, a sample with a lower abnormality rate (e.g., 20% abnormal sequence) is more prone to misclassification compared to a sample with 100% abnormal signals. To address this issue, we assign various weights to each abnormal sequence with a different abnormality ratio, therefore the learning model can identify abnormalities at an earlier stage than traditional methods. The rationale for this is that an algorithm has learned the underlying patterns within mixed instances by increasing their importance during training, enabling it to make more accurate predictions regarding the class to which a data sample belongs, even when the occurrence of abnormal signals is infrequent.

To achieve this, we propose a new weighting approach. Assume a mini-batch sample includes d^- normal and d^+ abnormal data points. Considering our data generation described in Section 2.1, assume that d_s^+ denote the number of abnormal signals with abnormality rates of η_s, where $s \in S = \{1,2,3,4,5\}$, $\eta_s \in \{0.2, 0.4, 0.6, 0.8, 1\}$, and $d^+ = \sum_{s\in S} d_s^+$. The standard method defines the weights of normal and abnormal data points as $\alpha^- = \frac{1}{d^-}$ and $\alpha^+ = \frac{1}{d^+}$, respectively. MRW gradually adjusts these weights to either *emphasize* or *de-emphasize* the importance of abnormal data points with higher abnormality rates. For this purpose, we emphasize the abnormal sequences with higher abnormality rates using $q_\lambda(\eta) = 1+\eta^\lambda$ (family of increasing functions). The parameter λ controls how weights of abnormal data points with various abnormality rates should be compared to each other.

Similarly, MRW approach de-emphasizes the importance of abnormal sequences with higher abnormality rates using a family of decreasing functions denoted as $q_\lambda(\eta) = 1+(1-\eta)^\lambda$. Fig. 5(a) and Fig. 5(b) demonstrate these two parametric curves for different values of λ. Therefore, the new weights for the abnormal data points with



(a) Validation G-mean and loss versus epochs (b) Validation G-mean versus validation loss

Fig. 7. Bi-objective early stopping. It includes monitoring both loss and G-mean values on the validation set in (a) three dimensions (versus the number of epochs), and (b) two dimensions. The red point represents the solution obtained through early stopping with validation loss, the yellow point represents the solution of early stopping with the validation G-mean, and the dark blue point indicates the solution derived from the bi-objective criteria.

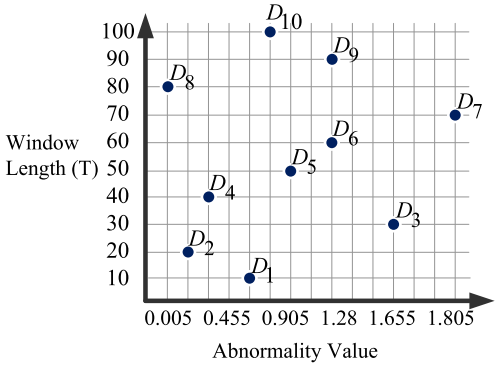


Fig. 8. Ten randomly selected data sets $(D_1, D_2, \ldots, D_{10})$ from all possible CCPR problems.

the abnormality rate η_s will be $\alpha_s^+ := q_\lambda(\eta_s) \times \frac{1}{d^+}$, $s \in S$ (Fig. 4). When the hyperparameter λ is equal to zero, the MRW will be similar to the standard weighting scheme as shown by the blue horizontal line in Figs. 5(a)–5(b). The adjusted weights for normal samples can be computed using the formula: $\alpha^- := \frac{1}{d^-}\left[1 - \sum_{s\in S}(\bar{q} - q_\lambda(\eta_s)) \times \frac{d_s^+}{d^+}\right]$ where $\bar{q} = \max_{s\in S} q_\lambda(\eta_s)$.

Remark 2. The MRW approach maintains the relative balance between the two classes denoted as $\alpha^- d^- = \sum_{s\in S} \alpha_s^+ d_s^+$.

2.4. Performance measures

In this study, we utilized two distinct sets of measures to assess the performance of the proposed algorithms: (a) data mining-based metrics, used for class imbalance problems, and (b) early detection-based metrics, which assess how quickly an algorithm can detect anomalous patterns in relation to abnormal samples.

2.4.1. Data mining-based evaluation metrics

For binary classification problems in CCPR, certain performance measures are used derived from a confusion matrix (Table 2). In particular, we report the accuracy, sensitivity, specificity, and G-mean to evaluate the models' performance as defined below,

Table 2
Binary confusion matrix.

		Actual	
		Abn*	N*
Predicted	Abn	True Positive (TP)	False Positive (FP)
	N	False Negative (FN)	True Negative (TN)

∗ Abn: Abnormal, N: Normal.

$$\begin{aligned}
\text{Accuracy} &= \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \\
\text{Sensitivity} &= \frac{\text{TP}}{\text{TP}+\text{FN}} \\
\text{Specificity} &= \frac{\text{TN}}{\text{TN}+\text{FP}} \\
\text{G-mean} &= \sqrt{\text{Sensitivity} \times \text{Specificity}}
\end{aligned}$$

2.4.2. Early detection-based metrics

In addition to measuring the performance of classifiers using the confusion matrix, it is necessary to measure their capability to quickly detect abnormal patterns. The average run length (ARL) is a common evaluation measure for CCPR problems. ARL is the expected number of samples required to identify an abnormal pattern (Knoth, 2006). However, recognizing that an abnormal pattern within a sequence might remain entirely undetected, the average run length index (ARLIDX) is introduced (Hwarng & Hubele, 1993b). ARLIDX measures the early detection capacity by calculating the ratio of ARL to the rate at which abnormal patterns are successfully detected. The computation of ARLIDX is based on the rolling window (RW) technique to systematically scan a data sequence for abnormal patterns (García, Peñabaena-Niebles, Jubiz-Diaz, & Perez-Tafur, 2022; Hwarng & Hubele, 1993a; Ünlü, 2021a). However, it is likely that an abnormal data sample may not be detected within the pre-defined detection horizon. Therefore, we introduce the percentage of undetected samples (USP) to assess the undetected samples. The goal is to achieve the lowest possible ARLIDX and USP to detect an anomaly.

When assessing a classifier's early detection performance using RW-based metrics, it is possible to encounter an inconsistent classifier that

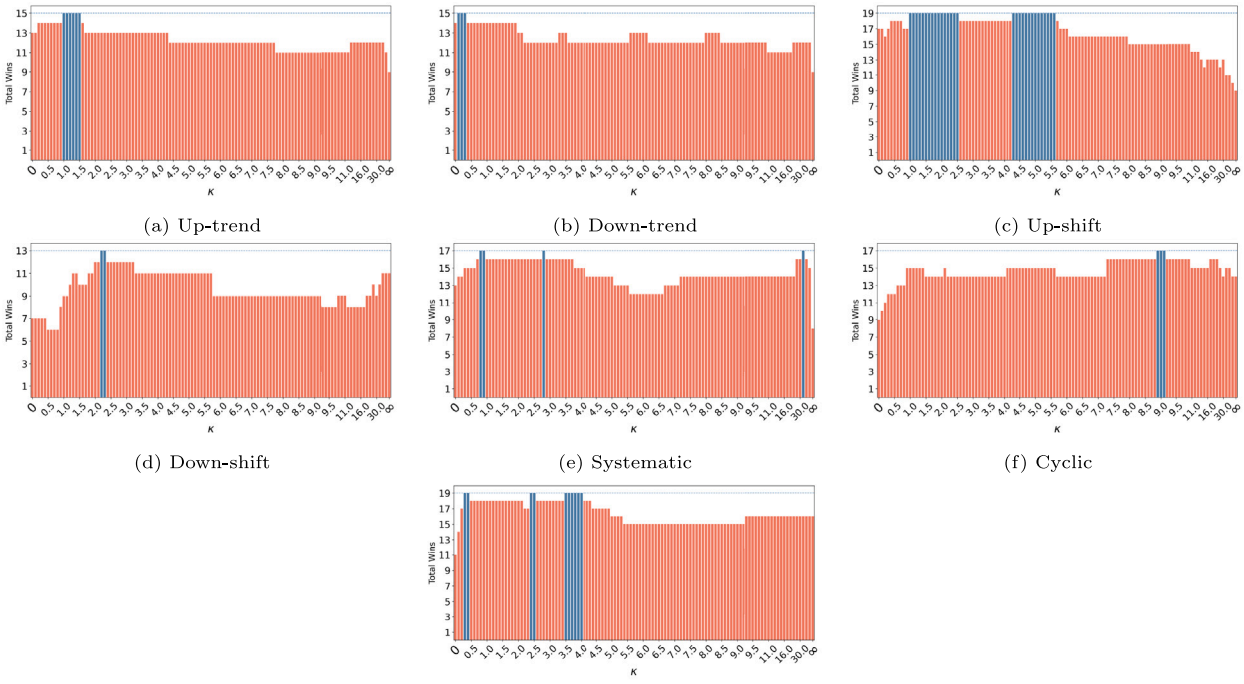


Fig. 9. Total number of wins cross 25 instances per abnormal pattern based on lower distance from the idea point. The best results with the highest number of wins are represented by the blue color. $\kappa = 0$ indicates LES method while $\kappa = \infty$ shows GES method.

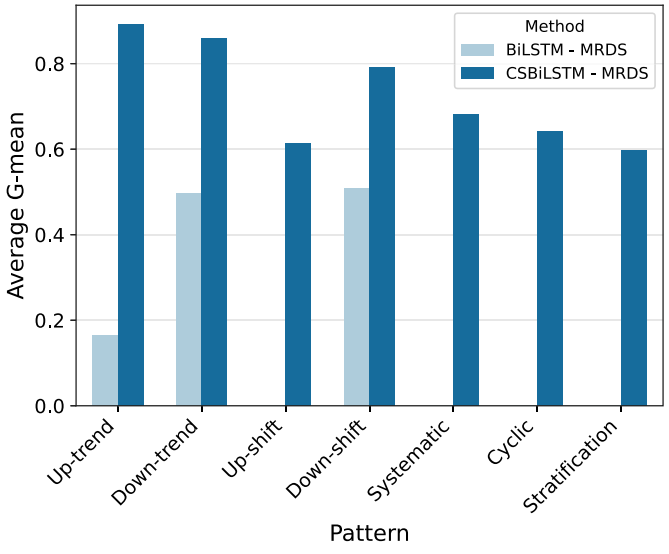


Fig. 10. Average G-mean scores of the selected models per pattern for CSBiLSTM - MRDS versus BiLSTM - MRDS.

alternates between true and false alerts, even with lower ARLIDX or USP values. Hence, we introduce a novel metric to evaluate the stability of a classifier in the early detection of abnormalities. This metric, defined as the true alert streaks rate from initial detection (TASRID), measures stability by determining the ratio of the maximum streaks of true alerts starting from the initial detection point to the total number of possible true detections. Fig. 6 shows two examples of the early detection performance of two classifiers on a rolled data point with a detection horizon of 8. In these figures, each circular point represents a detection attempt, where green "✓" denote true alerts and red "✘" marks indicate false alerts. TASRID measures the maximum number of consecutive true alerts starting from the initial detection point across all possible runs. In Fig. 6(a), where the classifier produces unstable predictions, TASRID is equal to $\frac{2}{8}$, whereas in Fig. 6(b), representing predictions with higher stability, TASRID is $\frac{6}{8}$. Remarkably, both cases exhibit similar ARLIDX and USP scores. TASRID values can range from 0 (indicating no successful detections in any of the given attempts) to 1 (indicating consistent true detections in all attempts).

2.4.3. A hybrid metric

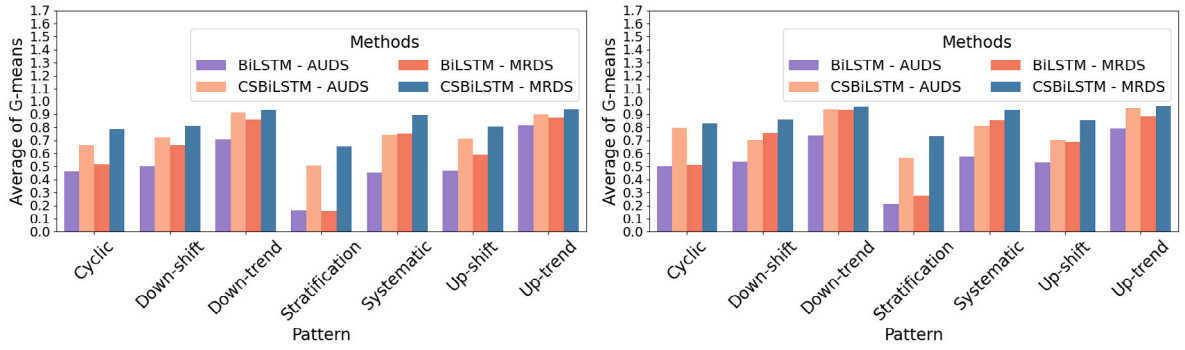
Comparing algorithms might be challenging when dealing with multiple performance metrics. To address this concern, we have combined the key measures such as G-mean, USP, ARLIDX, and TASRID into a single metric termed as the Unified Score defined as follows,

$$\text{Unified Score} := \sqrt[4]{\text{G-mean} \times (1 - \text{USP}) \times (1 - \frac{\text{ARLIDX}}{\text{H}}) \times \text{TASRID}} \tag{16}$$

where H represents the maximum possible ARLIDX value. An accurate, swift, and stable classifier should receive a unified score value close to 1.

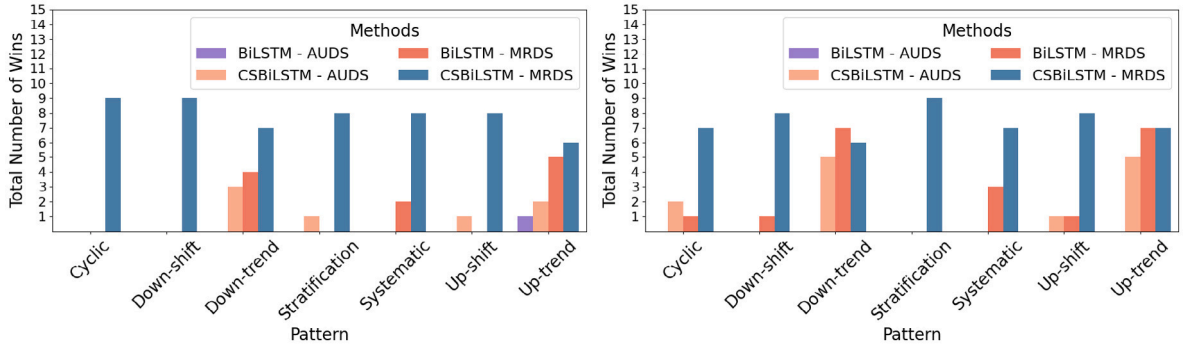
2.5. Bi-objective early stopping

Early stopping is a widely used strategy to determine the optimal point for terminating the training phase of a model. This method traditionally involves specifying a large number of epochs and tracking a chosen metric — usually the validation loss or the G-mean. Training is halted when there is no further improvement in this metric. It is crucial to note that the effectiveness of the early stopping technique is heavily influenced by the choice of the monitoring metric. The optimal number of epochs for the trained BiLSTM network is often obtained from the lowest (or highest) validation loss (or accuracy or G-mean). Minimizing loss and maximizing scores are closely related. However, it is typically advised against directly optimizing scores for training the



(a) T=50 (b) T=100

Fig. 11. Comparison of CSBiLSTM with BiLSTM using the MRDS and AUDS methods in terms of average G-mean.



(a) $T = 50$ (b) $T = 100$

Fig. 12. Comparison of CSBiLSTM with BiLSTM using the MRDS and AUDS methods in terms of number of wins.

neural network. This is due to the fact that the scores often represent a non-continuous function concerning the model's predictions (Marchetti et al., 2022). Moreover, the objectives of minimizing loss and maximizing G-mean often fail to coincide, especially in the context of highly imbalanced datasets. Hence, we introduce the Bi-objective Early Stopping (BES) method, specifically tailored to optimally balance between minimizing validation loss and maximizing validation G-mean in imbalanced datasets. This method employs a weighted combination of validation loss and G-mean as a basis to determine the optimal timing for updating the incumbent solution identified during training. We refer to the Loss Early Stopping as LES and G-mean Early Stopping as GES in this paper.

We define the loss percentage change (LPC) as a metric that measures the difference in the current validation loss, denoted as l, compared to the validation loss of the best solution achieved thus far, denoted as l^*. It is calculated using the formula: $LPC = \frac{l-l^*}{(l^*+\nu)}$, where ν represents a small constant used to prevent division by zero. Similarly, the G-mean percentage change (GPC) measures the change in the current validation G-mean value, denoted as G, compared to the G-mean of the best solution achieved thus far, denoted as G^*. It is calculated using the formula: $GPC = \frac{G-G^*}{(G^*+\nu)}$. At the end of each epoch, we calculate the weight update rule (WUR) measure defined by,

$$WUR = LPC - \kappa \times GPC \tag{17}$$

where $\kappa > 0$ is a hyperparameter that specifies the relative importance of improving the validation G-mean over compromising the validation loss. When $WUR < 0$, the best incumbent solution is updated; otherwise, we increase the patience number, which represents the number of epochs without improvement. Finally, the learning process is terminated if either the patience number or the number of epochs exceeds the given thresholds. Fig. 7 illustrates how this method selects the best number of epochs for the CSBiLSTM model during the training process. The training is conducted on an instance of systematic pattern with an abnormality value of $\xi_3 = 0.38$ and a window length of $T = 40$. The maximum number of epochs, the number of patience, and κ are set to 800, 120, and 2.5, respectively. In Fig. 7, each light blue point represents the classifier's performance as determined by the validation loss and G-mean.

As shown in Fig. 7(a), each point represents the evaluation of a BiLSTM network solution at a specific epoch, with green points indicating improvements to the current best incumbent solution. The LES method terminates training at epoch 118 and results in a validation loss and G-mean of 0.1609 and 0.706, respectively. In contrast, the GES method terminates training at epoch 26 and yields a validation loss and G-mean of 0.1678 and 0.7302, respectively (denoted by yellow point). Similarly, the BES method concludes at epoch 169, achieving validation loss and G-mean values of 0.1647 and 0.7263, respectively (denoted by red point).

Each green point in Fig. 7(b) corresponds to an update of the incumbent solution, numbered from 1 to 16. For the solutions related to updates 1 to 7, both LPC and GPC are improved. Consequently, $LPC < 0$ and $GPC > 0$, resulting in $WUR < 0$. However, at solution update 8, there is an improvement in G-mean, while the validation loss decreases. This indicates that $LPC > 0$ and $GPC > 0$. However, given that $\kappa = 2.5$ assigns higher importance to improving G-mean over compromising in validation loss, $WUR < 0$, leading to the recording of a new solution shown by number 8 in the figure. Similarly, at solution update 9, the optimizer discovers a solution with better validation loss but an inferior validation G-mean compared to solution 8. In this case, as improvement in loss outweighs the G-mean decline, $WUR < 0$, and solution 9 is recorded. The process of solution updates continues until the best solution is achieved in update 17, represented by the dark blue

point, which corresponds to epoch 169. Finally, the training process for this method is terminated at epoch 289 as the number of patience exceeds the given threshold of 120.

Remark 3. For a given solution with $WUR < 0$, there is an improvement in at least one of the validation loss or G-mean values.

Clearly, when κ gets a significantly large value, the bi-objective method closely mirrors the behavior of the GES approach. In such a situation, any improvement in validation loss is insufficient to offset even a minor decrease in G-mean. In contrast, when $\kappa = 0$, the BES method transforms into the LES method. In other words, the primary emphasis shifts to the validation loss, while the significance of G-mean improvement is disregarded.

2.6. Model selection

Fine-tuning the hyperparameters of the (CS-)BiLSTM models is crucial to achieving an increased performance for a DT equipped with CCPR. Table 3 summarizes a list of hyperparameters along with their corresponding sets of candidate values. Experimental results show that as the abnormality value in a problem instance decreases, the CCPR problem tends to become more challenging. Furthermore, the data collected in various industries may have various window lengths.

The standard hyperparameter tuning methods, including Bayesian optimization (BO) techniques (Bergstra, Bardenet, Bengio, & Kégl, 2011; Hutter, Hoos, & Leyton-Brown, 2011; Snoek, Larochelle, & Adams, 2012), are typically employed in machine learning domains. However, in the DT-assisted CCPR, where window lengths and abnormality rates vary, the standard methods cannot be directly employed. Thus, they may fail to suggest a predictive model capable of not only accommodating datasets with different window lengths but also performing effectively across all potential levels of abnormality values.

We present a tailored BO-based solution to this challenge. Instead of fine-tuning hyperparameters individually for every combination of window length and abnormality value, we initially selected $K = 10$ problem instances with diverse window lengths ($T \in [10, 100]$) and abnormality values ($\xi_1, \xi_2, \xi_3, \xi_4, \xi_5 \in [0.005, 1.805]$) for each abnormal pattern, as depicted in Fig. 8. It is worth mentioning that these ten problems were chosen from a single abnormal pattern (e.g., up-trend) and served as the baseline instances for model selection. Subsequently, we categorized the hyperparameters into two groups: the shared hyperparameters and the window-length specific (WLS) hyperparameters.

The shared hyperparameters are the number of hidden layers (L), learning rate (ρ), MRW approach, λ, and κ, while the WLS hyperparameters comprise the number of neurons per hidden layer ($N_1, N_2, \ldots, N_L$), window length (T), and dropout rates (δ) (see, Table 3). The shared hyperparameters are common among all selected window lengths, while the WLS parameters are tuned based on the selected window lengths. The details of the modified BO-based model selection approach are presented in Algorithm 1. The average of G-means of WLS models is reported as the overall score of the explored shared and WLS hyperparameters. Notably, increasing the number of CCPR problem instances can enhance the algorithm's ability to effectively approximate the best global model.

The categorization of hyperparameters into shared and WLS subsets helps to address the inherent complexity of CCPR problems. This approach is motivated by our understanding of the complex relation between the (CS)BiLSTM performance and the dynamic data streams in CCPR, as illustrated in Figs. 13–15. Our computational results show that both BiLSTM and CSBiLSTM models for each pattern are insensitive to variations in the number of hidden layers and learning rates. Therefore, we keep it the same across various window lengths and abnormality values for each pattern (shared parameters). In contrast, the WLS hyperparameters adapt to the dynamic nature of CCPR data, accommodating varying sequence lengths. Customizing hyperparameters such as the number of neurons and dropout rates ensures the model's optimal performance across various window lengths and abnormality values for each pattern.

Table 3
Summary of hyperparameter range for the proposed (CS-)BiLSTM.

Hyperparameter name	Range
Learning Rate (ρ)	$[0.0001, 0.0007, \ldots, 0.01]$
κ	$[0.01, 0.02, 0.03, \ldots, 50.0]$
MRW Approach**	[Emphasis, De-emphasis]
λ**	$[0, 0.1, 0.25, 0.33, 0.5, 1, 2, 3, 4, 10]$
No. Hidden Layers (L)	[2, 3, 4, 5]
Dropout Rates (δ)	$[0.1, 0.2, \ldots, 0.6]$
No. Neurons at Layer 1 (N_1)	$[T - R^*, T - R + 1, \ldots, T]$
No. Neurons at Layer 2 (N_2)	$[T - 2R, T - 2R + 1, \ldots, T - R]$
	...
No. Neurons at Layer L (N_L)	$[T - L \times R, T - L \times R + 1, \ldots, T - (L-1) \times R]$

*: $R = \lceil \frac{T-2}{L} \rceil$.
** : These hyperparameters are only applied for the CSBiLSTM.

Algorithm 1: The proposed BO-based model selection approach for the CCPR problem using the (CS-)BiLSTM algorithm.

Input:
- The number of datasets, denoted as K (with a default value of 10), each comprising normalized training sets (D_k^{TRN}) and validation sets (D_k^{VAL}), with an associated window length (T_k) for $k = 1, 2, \ldots, K$.
- Number of initial points (N^{IP})
- Number of tuning iterations (N^{TI})

Initialization:
$\mathbf{G} = [\,]$ {List $\mathbf{G}$ includes the G-mean of evaluated models}
$\mathbf{H} = [\,]$ {List $\mathbf{H}$ includes the explored hyperparameters}
$\mathbf{M} = [\,]$ {List $\mathbf{M}$ includes the WLS for (CS-)BiLSTM models}
$\mathbf{S} = [\,]$ {List $\mathbf{S}$ includes the score of explored hyperparameters}

for $i = 1 : N^{IP}$ **do**
  $\mathbf{H}_i^S \leftarrow$ Randomly select the shared hyperparameters
  for $k = 1 : K$ **do**
    $\mathbf{H}_{(i,k)}^{WLS} \leftarrow$ Randomly select the WLS hyperparameters conditioned on $\mathbf{H}_i^S$ and T_k
    $\mathbf{M}_{(i,k)} \leftarrow$ Build the (CS-)BiLSTM model using $\mathbf{H}_i^S$ and $\mathbf{H}_{(i,k)}^{WLS}$
    $\mathbf{G}_{(i,k)} \leftarrow$ Train $\mathbf{M}_{(i,k)}$ on D_k^{TRN} employing BES and evaluate on D_k^{VAL} for validation G-mean
  end for
  $\mathbf{S}_i \leftarrow$ Append $\sqrt[K]{\prod_{k=1}^{K} \mathbf{G}_{(i,k)}}$ as the score of hyperparameters $\mathbf{H}_i = (\mathbf{H}_i^S, \mathbf{H}_{(i,1)}^{WLS}, \ldots, \mathbf{H}_{(i,K)}^{WLS})$
end for

Tuning:
for $i = N^{IP} + 1 : N^{IP} + N^{TI}$ **do**
  $\mathbf{H}_i = (\mathbf{H}_i^S, \mathbf{H}_{(i,1)}^{WLS}, \ldots, \mathbf{H}_{(i,K)}^{WLS}) \leftarrow$ Apply Gaussian process regression using $\mathbf{H}$ and $\mathbf{S}$
  for $k = 1 : K$ **do**
    $\mathbf{M}_{(i,k)} \leftarrow$ Build the (CS-)BiLSTM model using $\mathbf{H}_i^S$ and $\mathbf{H}_{(i,k)}^{WLS}$
    $\mathbf{G}_{(i,k)} \leftarrow$ Train $\mathbf{M}_{(i,k)}$ on D_k^{TRN} employing BES and evaluate on D_k^{VAL} for validation G-mean
  end for
  $\mathbf{S}_i \leftarrow$ Append $\sqrt[K]{\prod_{k=1}^{K} \mathbf{G}_{(i,k)}}$ as the score of $\mathbf{H}_i$
end for
$i^* \leftarrow \arg\max_i \mathbf{S}_i$
Output: The best set of hyperparameters $\mathbf{H}_{i^*}$

3. Results and discussion

In this section, we present experimental results of the proposed CS-BiLSTM on various sets of simulated control chart datasets, along with a real-world dataset obtained from a wafer manufacturing industry (Chen

Table 4
Summary of parameter range for computational experiments.

Pattern	Window length	Abnormal parameter range
Up-trend	20	$\xi_1 \in [0.005, 1.805]$
Down-trend	20	$\xi_1 \in [0.005, 1.805]$
Up-shift	30	$\xi_2 \in [0.005, 1.805]$
Down-shift	30	$\xi_2 \in [0.005, 1.805]$
Systematic	70	$\xi_3 \in [0.005, 1.805]$
Cyclic	70	$\xi_4 \in [0.005, 1.805]$
Stratification	50	$\xi_5 \in [0.005, 1.805]$

et al., 2015) and a biomanufacturing industry. We evaluate the performance of these algorithms using the evaluation metrics introduced in Section 2.4. The BiLSTM and CSBiLSTM algorithms are implemented in Python version 3.9.7 with the Keras 2.6 framework (Chollet et al., 2015) and TensorFlow 2.6 libraries (Martín, Ashish, Paul, Eugene, Zhifeng, Craig, Greg, Andy, Jeffrey, Matthieu, Sanjay, Ian, Andrew, Geoffrey, Michael, Yangqing, Rafal, Lukasz, Manjunath, Josh, Dandelion, Rajat, Sherry, Derek, Chris, Mike, Jonathon, Benoit, Ilya, Kunal, Paul, Vincent, Vijay, Fernanda, Oriol, Pete, Martin, Martin, Yuan, & Xiaoqiang, 2015). Our experiments were conducted on an established OU Supercomputing Center for Education and Research (OSCER).

For each classification problem, a total of 5000 data samples are generated. We use the symbol ζ to represent the imbalanced class ratio, defined as the ratio of the number of normal samples (n^-) to the total number of samples ($n^+ + n^-$), expressed as $\zeta = \frac{n^-}{(n^+ + n^-)}$. In our experiments, 60% data is used for training, 20% is used as a validation set, and 20% is used as a test set. Prior to classification, all data are normalized to ensure they have a mean of zero and a standard deviation of one.

3.1. Performance analysis of bi-objective early stopping

We have compared the performance of our developed BES with the LES and GES monitoring techniques. To ensure representative results, we have carefully selected 25 instances for each abnormal pattern, incorporating diverse abnormal values within the range $[0.005, 1.805]$ for a fixed window length, as shown in Table 4. For the experiments, we set the number of patience to 200, and the maximum number of epochs to 1500.

Since the parameter κ considerably affects the performance of our proposed method, we perform extensive computation experiments using a wide range of values from 0.01 to 100. We note that $\kappa = 0$ represents LES method and $\kappa = \infty$ indicates GES method. We train a similar BiLSTM model on the training set for a considerable number of epochs, while simultaneously applying the early stopping methods to the validation set. Training is terminated once all methods indicate the need for early stopping.

To evaluate the performance of each early stopping method, we measure the frequency at which the pairs of loss and G-mean (ℓ_k, G_k), generated by these methods for $k = 1$ to 25, achieve the minimum Euclidean distance to the ideal point, characterized by zero loss and a G-mean of one. A method earns a win whenever its calculated distance, $\sqrt{\ell_k^2 + (G_k - 1)^2}$, is the lowest compared to the distances calculated for other methods. Fig. 9 illustrates these comparisons for each pattern over various κ values. Notably, certain patterns, such as trend and shift, demonstrate a higher number of wins at lower κ values, suggesting a leaning toward loss minimization in achieving proximity to the ideal point. On the contrary, patterns such as cyclic peak at higher κ values, indicating a favorable shift toward G-mean maximization. This observation aligns with prior research that optimizing complex patterns, e.g., cyclic, may not significantly improve loss minimization. Rather, focusing on improving G-mean can increase the chance of obtaining solutions that are closer to the ideal point.

Table 5
Pairwise Wilcoxon signed-rank tests for BES, LES, and GES across patterns.

Pattern	Comparison	Δ rank ($p-$value)
Up-trend	BES vs. GES	59(<0.05*)
	BES vs. LES	0(<0.05*)
	GES vs. LES	118(0.361)
Down-trend	BES vs. GES	48(<0.05*)
	BES vs. LES	0(<0.05*)
	GES vs. LES	137(0.710)
Up-shift	BES vs. GES	10(<0.01**)
	BES vs. LES	0(<0.05*)
	GES vs. LES	41(<0.05*)
Down-shift	BES vs. GES	14(<0.01**)
	BES vs. LES	0(<0.001***)
	GES vs. LES	130(0.396)
Systematic	BES vs. GES	14(<0.01**)
	BES vs. LES	0(<0.01**)
	GES vs. LES	133(0.879)
Cyclic	BES vs. GES	7(<0.05*)
	BES vs. LES	20(<0.01**)
	GES vs. LES	111(0.265)
Stratification	BES vs. GES	0(<0.05*)
	BES vs. LES	1(<0.001***)
	GES vs. LES	41(0.053)

Furthermore, we compare the performance of our method against LES and GES methods using pairwise Wilcoxon rank sum test. Table 5 details these comparisons. For most patterns, our method consistently achieves statistically significant smaller distances to the ideal point compared to the LES and GES methods, highlighting its superior performance.

3.2. Best selected models

To enhance the performance of our BiLSTM models, we utilize Algorithm 1, which includes an extensive set of hyperparameters listed in Table 3. The models selected for each pattern using data generated by MRDS are summarized in Tables 6 and 7 for CSBiLSTM and BiLSTM, respectively. Fig. 10 presents the average G-mean score for CSBiLSTM versus BiLSTM for each abnormal pattern based on the model selection described in Algorithm 1. The average G-mean values for CSBiLSTM models are consistently higher than those for BiLSTM across all patterns.

3.3. Comparative analysis of the synthetic data generation method

We have conducted a comparative analysis between the conventional data generation method (AUDS) and the MRDS. To achieve this, we have transformed the MRDS data into five distinct mix ratios of AUDS data, each representing varying levels of abnormality rates defined earlier (20%, 40%, 60%, 80%, and 100%). For instance, when dealing with sequences containing 20% abnormal signals within a window length of $T = 100$, we obtain conventional data with a reduced sequence length of 20, consisting of only the last 20% of abnormal signals. Similarly, we follow this procedure for other abnormality rates, resulting in truncated sequences with lengths of $40, 60, 80$, and 100.

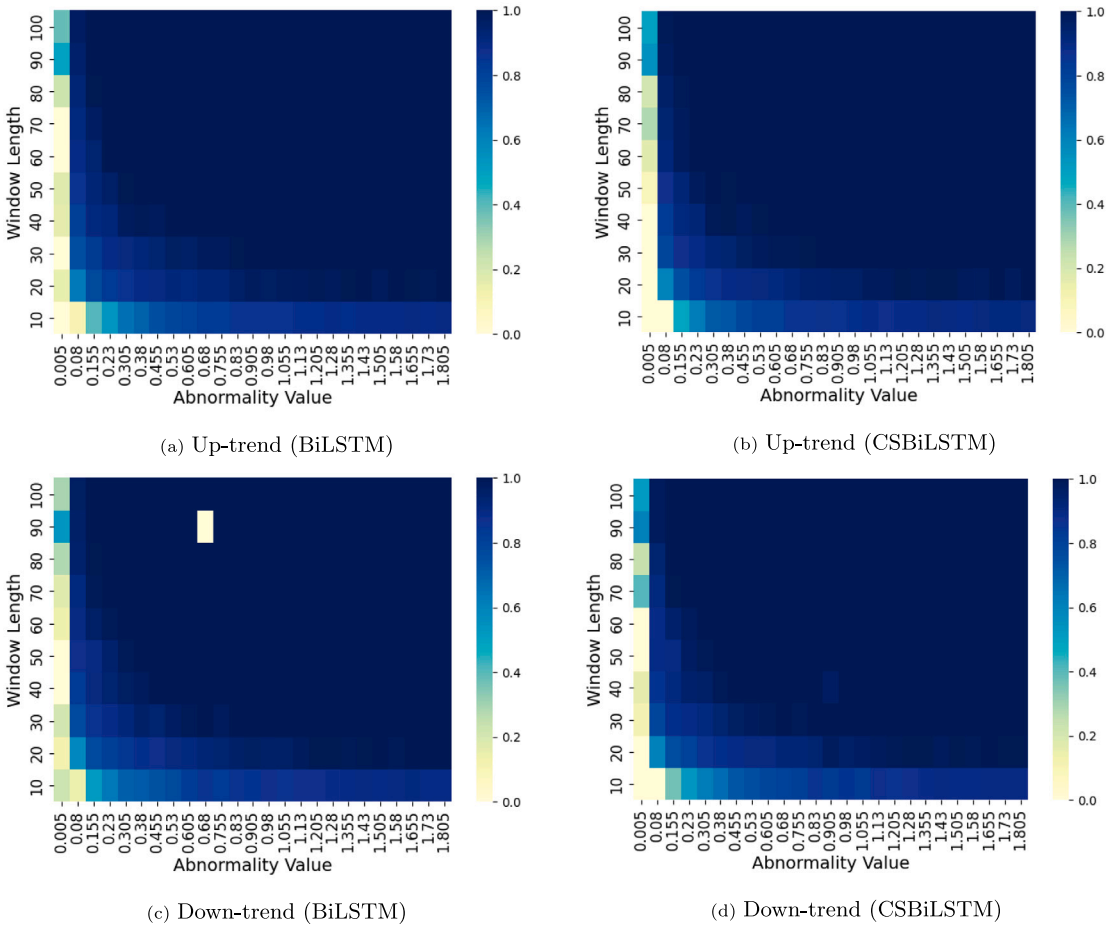
Our training dataset comprises 3000 data samples, with a total of 150 abnormal samples. These abnormal samples are distributed equally, with 30 samples for each abnormality rate. Similarly, in the validation dataset, there are 50 abnormal samples, distributed as 10 samples for each rate. This leads to an imbalanced ratio of 95% in both sets. For each of the obtained five AUDS subsets, we train and evaluate both traditional BiLSTM and CSBiLSTM classifiers using the corresponding data points, and the results are combined to obtain the overall G-mean. This overall G-mean can be directly compared to the G-mean obtained when the same BiLSTM and CSBiLSTM are trained using

Table 6
The selected hyperparameters of the CSBiLSTM method for each pattern. N_l and δ_l denote the number of neurons and dropout rate in layer l.

Pattern (ρ, κ, MRW, λ)	Layer (l)	(N_l, δ_l) for various window lengths ($T \in \{10, 20, \ldots, 100\}$)									
		10	20	30	40	50	60	70	80	90	100
Up-trend (6.1, 11, D, 0.1)	1	(9, 0.5)	(14, 0.5)	(23, 0.5)	(32, 0.5)	(39, 0.5)	(51, 0.5)	(51, 0.5)	(69, 0.5)	(88, 0.5)	(83, 0.5)
	2	(6, 0.5)	(13, 0.5)	(15, 0.5)	(19, 0.5)	(27, 0.5)	(38, 0.5)	(35, 0.5)	(48, 0.5)	(32, 0.5)	(64, 0.5)
	3	(5, 0.5)	(2, 0.5)	(10, 0.5)	(9, 0.5)	(17, 0.5)	(4, 0.5)	(13, 0.5)	(24, 0.5)	(25, 0.5)	(4, 0.5)
Down-trend (6.7, 14.4, E, 4)	1	(9, 0.3)	(19, 0.4)	(22, 0.2)	35, 0.1)	(35, 0.3)	(58, 0.6)	(66, 0.3)	(70, 0.4)	(84, 0.3)	(92, 0.3)
	2	(8, 0.4)	(12, 0)	(16, 0)	(26, 0.4)	(30, 5)	(34, 0.3)	(30, 0.1)	(43, 0.2)	(51, 0.4)	(47, 0.2)
	3	(4, 0.5)	(5, 0.4)	(5, 0.2)	(15, 0.5)	(10, 0.3)	(10, 0.4)	(25, 0.5)	(10, 0.3)	(25, 0.5)	(31, 0)
Up-shift (4.9, 9.6, E, 0.25)	1	(9, 0.3)	(19, 0)	(26, 0)	(36, 0.1)	(48, 0.5)	(53, 0.5)	(49, 0.1)	(73, 0.2)	(87, 0.2)	(98, 0.3)
	2	(6, 0.2)	(8, 0)	(12, 0.4)	(20, 0.1)	(32, 0.1)	(30, 0.5)	(27, 0.2)	(30, 0.1)	(33, 0.5)	(49, 0.5)
	3	(6, 0.2)	(4, 0.3)	(10, 0.3)	(11, 0.5)	(15, 0.2)	(14, 0.2)	(17, 0.5)	(21, 0.4)	(30, 0.4)	(23, 0.5)
Down-shift (3.1, 12.1, D, 1)	1	(10, 0.3)	(17, 0.6)	(29, 0.3)	(35, 0.2)	(46, 0.4)	(57, 0.1)	(60, 0.3)	(72, 0.2)	(76, 0.2)	(92, 0.2)
	2	(8, 0.5)	(16, 0)	(24, 0.5)	(27, 0.4)	(39, 0.4)	(46, 0.5)	(50, 0.2)	(41, 0.1)	(61, 0.4)	(75, 0.4)
	3	(7, 0.2)	(11, 0.6)	(16, 0.4)	(23, 0.4)	(25, 0.2)	(31, 0.4)	(33, 0.2)	(50, 0.4)	(47, 0.4)	(59, 0.4)
	4	(6, 0.6)	(8, 0.5)	(14, 0.6)	(19, 0.5)	(23, 0.5)	(24, 0.2)	(30, 0.5)	(31, 0.3)	(34, 0.3)	(31, 0.3)
	5	(6, 0.3)	(5, 0.3)	(8, 0.3)	(6, 0.4)	(9, 0.3)	(12, 0.4)	(6, 0.5)	(16, 0.1)	(8, 0.4)	(20, 0.2)
Systematic (8.5, 2.5, D, 4)	1	(6, 0.5)	(19, 0.4)	(29, 0)	(30, 0.3)	(40, 0.1)	(47, 0.4)	(64, 0.6)	(42, 0.4)	(76, 0.2)	(83, 0.2)
	2	(3, 0.6)	(10, 0.4)	(13, 0.2)	(9, 0.4)	(13, 0)	(10, 0.5)	(3, 0.4)	(40, 0.6)	(8, 0.5)	(14, 0.5)
Cyclic (3.1, 2.8, D, 0.33)	1	(9, 0.5)	(18, 0.5)	(26, 0.5)	(36, 0.5)	(44, 0.5)	(53, 0.5)	(62, 0.5)	(70, 0.5)	(79, 0.5)	(88, 0.5)
	2	(7, 0.5)	(14, 0.5)	(20, 0.5)	(26, 0.5)	32, 0.5)	(39, 0.5)	(44, 0.5)	(52, 0.5)	(57, 0.5)	(64, 0.5)
	3	(5, 0.5)	(10, 0.5)	(12, 0.5)	(18, 0.5)	(20, 0.5)	(25, 0.5)	(28, 0.5)	(32, 0.5)	(35, 0.5)	(40, 0.5)
	4	(3, 0.5)	(6, 0.5)	(6, 0.5)	(8, 0.5)	(8, 0.5)	(11, 0.5)	(10, 0.5)	(14, 0.5)	(13, 0.5)	(16, 0.5)
Stratification (6.7, 10.6, D, 10)	1	(10, 0.6)	(18, 0.4)	(29, 0.4)	(38, 0.3)	(49, 0.3)	(60, 0.1)	(68, 0.1)	(70, 0.1)	(79, 0)	(82, 0.4)
	2	(8, 0.3)	(15, 0.2)	(21, 0.1)	(28, 0.4)	(33, 0.1)	(41, 0)	(50, 0.2)	(55, 0.2)	(71, 0.5)	(74, 0.5)
	3	(8, 0.1)	(13, 0.4)	(18, 0.1)	(19, 0.2)	(24, 0)	(35, 0.6)	(41, 0)	(48, 0.2)	(55, 0.3)	(44, 0.4)
	4	(6, 0.2)	(8, 0)	(15, 0.4)	(14, 0.2)	(19, 0.6)	(21, 0.1)	(25, 0.3)	(26, 0.6)	(35, 0.4)	(28, 0.1)
	5	(6, 0.5)	(6, 0.1)	(9, 0.4)	(7, 0.4)	(14, 0.4)	(15, 0.4)	(16, 0.5)	(17, 0.3)	(15, 0.4)	(8, 0)

Table 7
The selected hyperparameters of the BiLSTM method for each pattern.

Pattern ($\rho(\times 10^{-3}), \kappa$)	Layer (l)	(N_l, δ_l) for various window lengths ($T \in \{10, 20, \ldots, 100\}$)									
		10	20	30	40	50	60	70	80	90	100
Up-trend (9.7, 3.4)	1	(9, 0.5)	(18, 0.5)	(28, 0.5)	(31, 0.5)	(42, 0.5)	(57, 0.5)	(59, 0.5)	(59, 0.5)	(72, 0.5)	(96, 0.5)
	2	(7, 0.5)	(9, 0.5)	(15, 0.5)	(26, 0.5)	(26, 0.5)	(39, 0.5)	(47, 0.5)	(35, 0.5)	(47, 0.5)	(54, 0.5)
	3	(6, 0.5)	(4, 0.5)	(7, 0.5)	(6, 0.5)	(11, 0.5)	(21, 0.5)	(10, 0.5)	(26, 0.5)	(13, 0.5)	(9, 0.5)
Down-trend (7.8, 6.3)	1	(9, 0.0)	(19, 0.3)	(25, 0.2)	(36, 0.1)	(49, 0.5)	(50, 0.2)	(62, 0.3)	(66, 0.1)	(81, 0.2)	(87, 0.6)
	2	(8, 0.2)	(15, 0.0)	(20, 0.0)	(33, 0.3)	(39, 0.4)	(45, 0.2)	(45, 0.4)	(61, 0.4)	(58, 0.3)	(80, 0.3)
	3	(8, 0.3)	(12, 0.1)	(16, 0.0)	(21, 0.0)	(23, 0.4)	(37, 0.5)	(41, 0.2)	(46, 0.1)	(50, 0.1)	(62, 0.0)
	4	(7, 0.3)	(11, 0.1)	(12, 0.2)	(16, 0.1)	(16, 0.2)	(17, 0.4)	(26, 0.5)	(25, 0.0)	(29, 0.2)	(29, 0.1)
	5	(6, 0.6)	(7, 0.5)	(6, 0.0)	(6, 0.5)	(14, 0.4)	(8, 0.1)	(14, 0.1)	(19, 0.5)	(15, 0.5)	(23, 0.2)
Up-shift (3.1, 2.6)	1	(9, 0.0)	(18, 0.5)	(28, 0.5)	(36, 0.5)	(46, 0.5)	(54, 0.5)	(64, 0.5)	(72, 0.5)	(82, 0.5)	(90, 0.5)
	2	(9, 0.4)	(16, 0.5)	(22, 0.5)	(30, 0.5)	(36, 0.5)	(44, 0.5)	(50, 0.5)	(58, 0.5)	(64, 0.5)	(72, 0.5)
	3	(7, 0.2)	(12, 0.5)	(18, 0.5)	(22, 0.5)	(28, 0.5)	(32, 0.5)	(38, 0.5)	(42, 0.5)	(48, 0.5)	(52, 0.5)
	4	(6, 0.5)	(10, 0.5)	(12, 0.5)	(16, 0.5)	(18, 0.5)	(22, 0.5)	(24, 0.5)	(28, 0.5)	(30, 0.5)	(34, 0.5)
	5	(5, 0.3)	(6, 0.5)	(8, 0.5)	(8, 0.5)	(10, 0.5)	(10, 0.5)	(12, 0.5)	(12, 0.5)	(14, 0.5)	(14, 0.5)
Down-shift (3.1, 5.3)	1	(10, 0.0)	(20, 0.0)	(29, 0.1)	(35, 0.1)	(45, 0.1)	(56, 0.3)	(59, 0.1)	(76, 0.5)	(63, 0.3)	(78, 0.0)
	2	(7, 0.2)	(12, 0.3)	(17, 0.4)	(26, 0.1)	(34, 0.1)	(28, 0.4)	(36, 0.4)	(29, 0.2)	(59, 0.6)	(37, 0.2)
	3	(6, 0.2)	(4, 0.1)	(7, 0.6)	(14, 0.3)	(14, 0.2)	(20, 0.0)	(12, 0.3)	(6, 0.1)	(28, 0.5)	(5, 0.5)
Systematic (2.5, 2.6)	1	(9, 0.5)	(18, 0.5)	(26, 0.5)	(36, 0.5)	(44, 0.5)	(53, 0.5)	(62, 0.5)	(70, 0.5)	(79, 0.5)	(88, 0.5)
	2	(7, 0.5)	(14, 0.5)	(20, 0.5)	(26, 0.5)	(32, 0.5)	(39, 0.5)	(44, 0.5)	(52, 0.5)	(57, 0.5)	(64, 0.5)
	3	(5, 0.5)	(10, 0.5)	(12, 0.5)	(18, 0.5)	(20, 0.5)	(25, 0.5)	(28, 0.5)	(32, 0.5)	(35, 0.5)	(40, 0.5)
	4	(3, 0.5)	(6, 0.5)	(6, 0.5)	(8, 0.5)	(8, 0.5)	(11, 0.5)	(10, 0.5)	(14, 0.5)	(13, 0.5)	(16, 0.5)
Cyclic (6.7, 2.6)	1	(9, 0.5)	(18, 0.5)	(26, 0.5)	(36, 0.5)	(44, 0.5)	(53, 0.5)	(62, 0.5)	(70, 0.5)	(79, 0.5)	(88, 0.5)
	2	(7, 0.5)	(14, 0.5)	(20, 0.5)	(26, 0.5)	(32, 0.5)	(39, 0.5)	(44, 0.5)	(52, 0.5)	(57, 0.5)	(64, 0.5)
	3	(5, 0.5)	(10, 0.5)	(12, 0.5)	(18, 0.5)	(20, 0.5)	(25, 0.5)	(28, 0.5)	(32, 0.5)	(35, 0.5)	(40, 0.5)
	4	(3, 0.5)	(6, 0.5)	(6, 0.5)	(8, 0.5)	(8, 0.5)	(11, 0.5)	(10, 0.5)	(14, 0.5)	(13, 0.5)	(16, 0.5)
Stratification (5.5, 2.6)	1	(9, 0.5)	(18, 0.5)	(26, 0.5)	(36, 0.5)	(44, 0.5)	(53, 0.5)	(62, 0.5)	(70, 0.5)	(79, 0.5)	(88, 0.5)
	2	(7, 0.5)	(14, 0.5)	(20, 0.5)	(26, 0.5)	(32, 0.5)	(39, 0.5)	(44, 0.5)	(52, 0.5)	(57, 0.5)	(64, 0.5)
	3	(5, 0.5)	(10, 0.5)	(12, 0.5)	(18, 0.5)	(20, 0.5)	(25, 0.5)	(28, 0.5)	(32, 0.5)	(35, 0.5)	(40, 0.5)



(a) Up-trend (BiLSTM)

(b) Up-trend (CSBiLSTM)

(c) Down-trend (BiLSTM)

(d) Down-trend (CSBiLSTM)

Fig. 13. Obtained boundary for inseparable, partially separable, and separable classification for BiLSTM and CSBiLSTM methods in up-trend and down-trend patterns.

the MRDS dataset. For simplicity, we have designated the traditional BiLSTM using both the conventional data generation method and MRDS methods as BiLSTM-AUDS and BiLSTM-MRDS, respectively. Similarly, the CSBiLSTM models utilizing the AUDS and MRDS methods are represented as CSBiLSTM-AUDS and CSBiLSTM-MRDS, respectively. The results are illustrated in Figs. 11 through 12. Further, we implement this approach to various abnormal patterns with two window lengths of 50 and 100 as well as nine problem instances representing various levels of problem difficulty. The average of G-means is obtained from these nine problem instances. The plots show that the training of a CSBiLSTM using the MRDS approach outperforms the other methods in all patterns. Moreover, Fig. 12 shows the number of times each method wins across the nine problem instances. These results demonstrate the superiority of the CSBiLSTM trained with MRDS data, as it consistently performs better in terms of G-mean values compared to the other approaches. In particular, the CSBiLSTM with MRDS results into a superior performance compared to the conventional method in all patterns as shown in Fig. 11.

3.4. Comparative performance heatmaps

We conduct a comparative analysis of CSBiLSTM and BiLSTM for each abnormal pattern using MRDS, considering various window lengths (T) and different pattern parameters, with a specific focus on G-mean. Figs. 13–15 show the results with highly imbalanced datasets where 95% of the data belongs to the normal class, and only 5% belongs to the abnormal class. In this experiment, we aim to identify the parameter values for which, (a) problems fully separable (FS) for both algorithms, (b) inseparable (IS) problems that cannot be solved by either algorithm, and (c) the remaining scenarios, characterized as partially separable (PS) problems. Based on the G-means in Figs. 13–15, we observe that the CCPR problem becomes easily solvable as the window length and the abnormality value increase. However, when the window length and abnormality value parameters are smaller, it often results in more complex detection problems. This finding aligns with recent research (Ünlü, 2021a; Xanthopoulos & Razzaghi, 2014).

In Figs. 13–15, one can distinguish the IS, PS, and FS regions obtained by the BiLSTM and CSBiLSTM algorithms for highly imbalanced datasets. Based on these figures, the BiLSTM and CSBiLSTM algorithms yield comparable performance measures for the up-trend and down-trend abnormal patterns. For the up-shift and down-shift patterns, the CSBiLSTM algorithm demonstrates superior performance with smaller window lengths, although it generally outperforms BiLSTM slightly. The primary strength of CSBiLSTM lies in its ability to yield higher G-mean values, particularly in non-trend patterns such as up-shift, down-shift, systematic, cyclic, and stratification.

The comparison between BiLSTM and CSBiLSTM models, shown in Fig. 16, provides interesting insights based on their win percentages. The CSBiLSTM demonstrates distinct strengths compared to BiLSTM for various abnormal patterns. For the up-shift, down-shift, systematic, cyclic, and stratification patterns, CSBiLSTM outperforms BiLSTM, winning 76%, 60%, 77%, 56%, and 87% of instances out of 250, respectively, in terms of achieving a higher G-mean. However, in the up-trend and down-trend patterns, both algorithms produce comparable results, with CSBiLSTM achieving wins percentages of 27% and 17%, while BiLSTM achieves 9% and 21% wins, respectively.

Table 8
Percentage of undetected samples for the up-shift pattern. ζ is the class imbalance percentage ratio.

ζ	T	ξ_2	AUDS		MRDS	
			BiLSTM	CSBiLSTM	BiLSTM	CSBiLSTM
95%	30	0.005	2.0	0.0	6.0	0.0
		0.08	23.3	0.0	5.7	0.0
		0.155	0.7	0.0	2.0	0.0
		0.23	7.7	0.0	1.0	0.0
		0.305	0.7	0.0	0.0	0.0
		0.38	0.7	0.0	0.0	0.0
		0.455	0.0	0.0	0.0	0.0
		0.53	0.0	0.0	0.0	0.0
		0.605	0.0	0.0	0.0	0.0
	50	0.005	100.0	0.0	29.3	0.0
		0.08	100.0	0.0	100.0	0.0
		0.155	55.7	0.0	100.0	0.0
		0.23	16.3	0.0	99.7	0.0
		0.305	3.3	0.0	12.3	0.0
		0.38	1.0	0.0	2.3	0.0
		0.455	0.0	0.0	0.0	0.0
		0.53	0.0	0.0	0.0	0.0
		0.605	0.0	0.0	0.0	0.0
	80	0.005	8.7	0.0	18.0	0.0
		0.08	97.7	1.3	12.3	0.0
		0.155	55.3	0.3	14.7	0.0
		0.23	12.3	1.0	10.7	0.0
		0.305	2.3	0.3	2.7	0.0
		0.38	0.7	0.0	0.3	0.0
		0.455	1.0	0.0	0.0	0.0
		0.53	0.7	0.3	0.0	0.0
		0.605	0.0	0.0	0.0	0.0
90%	30	0.005	1.3	0.0	0.0	0.0
		0.08	2.3	0.0	0.3	0.0
		0.155	0.0	0.0	0.0	0.0
		0.23	9.3	0.0	0.0	0.0
		0.305	0.0	0.0	0.7	0.0
		0.38	0.0	0.0	0.0	0.0
		0.455	0.0	0.0	0.0	0.0
		0.53	0.0	0.0	0.0	0.0
		0.605	0.0	0.0	0.0	0.0
	50	0.005	100.0	0.0	100.0	0.0
		0.08	100.0	0.0	100.0	0.0
		0.155	46.7	0.0	65.0	0.0
		0.23	8.3	0.0	24.0	0.0
		0.305	2.0	0.0	3.3	0.0
		0.38	0.0	0.0	0.3	0.0
		0.455	0.0	0.0	0.0	0.0
		0.53	0.0	0.0	0.0	0.0
		0.605	0.0	0.0	0.0	0.0
	80	0.005	3.3	0.0	45.7	0.0
		0.08	11.3	5.7	5.3	0.0
		0.155	11.7	2.7	7.0	0.3
		0.23	7.3	0.7	0.0	0.0
		0.305	1.0	0.0	0.7	0.0
		0.38	0.3	0.0	0.0	0.0
		0.455	0.0	0.0	0.0	0.0
		0.53	0.7	0.0	0.0	0.0
		0.605	0.3	0.0	0.0	0.0

Table 9
Percentage of undetected samples for the up-shift pattern (continued).

ζ	T	ξ_2	AUDS		MRDS	
			BiLSTM	CSBiLSTM	BiLSTM	CSBiLSTM
75%	30	0.005	0.0	0.0	0.0	0.0
		0.08	0.0	0.0	0.0	0.0
		0.155	3.3	0.0	0.0	0.0
		0.23	0.0	0.0	0.0	0.0
		0.305	0.0	0.0	0.0	0.0
		0.38	0.0	0.0	0.0	0.0
		0.455	0.0	0.0	0.0	0.0
		0.53	0.0	0.0	0.0	0.0
		0.605	0.0	0.0	0.0	0.0
	50	0.005	81.3	0.0	100.0	0.0
		0.08	0.3	0.0	82.7	0.0
		0.155	10.7	1.0	21.3	0.3
		0.23	0.7	0.0	3.3	0.0
		0.305	0.3	0.0	0.7	0.0
		0.38	0.0	0.0	0.0	0.0
		0.455	0.0	0.0	0.0	0.0
		0.53	0.0	0.0	0.0	0.0
		0.605	0.0	0.0	0.0	0.0
	80	0.005	0.0	0.0	0.0	0.0
		0.08	13.0	0.3	0.3	0.3
		0.155	5.3	1.7	9.0	0.3
		0.23	2.0	0.3	1.0	0.3
		0.305	1.3	0.7	0.7	0.0
		0.38	0.7	0.3	0.0	0.0
		0.455	0.0	0.0	0.0	0.0
		0.53	0.0	0.0	0.0	0.0
		0.605	0.0	0.0	0.0	0.0
50%	30	0.005	0.0	0.0	0.0	0.0
		0.08	0.0	0.0	0.0	0.0
		0.155	0.0	0.0	0.0	0.0
		0.23	0.0	0.0	0.0	0.0
		0.305	0.0	0.0	0.0	0.0
		0.38	0.0	0.0	0.0	0.0
		0.455	0.0	0.0	0.0	0.0
		0.53	0.0	0.0	0.0	0.0
		0.605	0.0	0.0	0.0	0.0
	50	0.005	0.0	0.0	0.0	0.0
		0.08	0.7	0.7	0.0	0.3
		0.155	0.3	0.7	0.0	0.0
		0.23	1.0	1.7	0.3	0.3
		0.305	0.0	0.0	0.0	0.0
		0.38	0.0	0.0	0.0	0.0
		0.455	0.0	0.0	0.0	0.0
		0.53	0.0	0.0	0.0	0.0
		0.605	0.0	0.0	0.0	0.0
	80	0.005	0.0	0.0	0.0	0.0
		0.08	3.7	1.3	0.0	0.0
		0.155	2.7	2.7	0.3	0.0
		0.23	0.7	0.3	0.3	0.0
		0.305	0.7	0.3	0.0	0.0
		0.38	0.3	0.3	0.0	0.0
		0.455	0.0	0.3	0.0	0.0
		0.53	0.0	0.0	0.0	0.0
		0.605	0.0	0.0	0.0	0.0

Table 10
Comparative analysis of our proposed CSBiLSTM (M) with Ünlü (2021a)'s method (U) using key performance metrics for up-trend, down-trend, and up-shift patterns. The best results are shown in bold font.

Pattern	T	ξ	G-mean		ARLIDX		USP		TASRID		Unified Score	
			M	U	M	U	M	U	M	U	M	U
Up-trend	30	0.305	0.96	0.77	5.36	15.62	0.0	0.0	0.75	0.88	**0.91**	0.88
		0.38	0.95	0.65	4.91	19.87	0.0	0.0	0.83	0.76	**0.93**	0.80
		0.455	0.96	0.63	5.68	22.36	0.0	0.0	0.96	0.80	**0.97**	0.80
		0.53	0.98	0.75	5.42	17.63	0.0	0.0	0.95	0.86	**0.97**	0.86
		0.605	1.00	0.85	5.19	11.90	0.0	0.0	0.96	0.91	**0.98**	0.91
		0.68	1.00	0.77	4.99	15.94	0.0	0.0	0.96	0.87	**0.98**	0.88
Down-trend	30	0.305	0.96	0.79	6.09	15.30	0.0	0.0	0.82	0.88	**0.93**	0.88
		0.38	0.92	0.72	4.46	18.00	0.0	0.0	0.85	0.86	**0.93**	0.85
		0.455	0.98	0.77	5.11	16.55	0.0	0.0	0.90	0.87	**0.96**	0.87
		0.53	0.99	0.79	4.82	16.84	0.0	0.0	0.90	0.87	**0.96**	0.88
		0.605	1.00	0.71	4.94	18.62	0.0	0.0	0.95	0.85	**0.98**	0.85
		0.68	0.98	0.65	6.09	19.60	0.0	0.0	0.96	0.85	**0.97**	0.82
Up-shift	50	0.305	0.73	0.73	9.95	21.54	0.0	0.0	0.26	0.38	0.55	**0.60**
		0.53	0.85	0.75	13.77	24.57	0.0	0.0	0.55	0.56	**0.81**	0.76
		0.755	0.92	0.75	8.68	27.44	0.0	0.0	0.71	0.70	**0.83**	0.82
		0.98	0.98	0.71	8.13	32.52	0.0	0.0	0.77	0.64	**0.93**	0.84
		1.205	0.98	0.72	7.58	29.35	0.0	0.0	0.79	0.69	**0.93**	0.81
		1.43	1.00	0.66	6.23	33.21	0.0	0.0	0.89	0.64	**0.94**	0.82

Table 11
Comparative analysis of our proposed CSBiLSTM (M) with Ünlü (2021a)'s method (U) using key performance metrics for down-shift, systematic, cyclic, and stratification patterns. The best results are shown in bold font.

Pattern	T	ξ	G-mean		ARLIDX		USP		TASRID		Unified score	
			M	U	M	U	M	U	M	U	M	U
Down-shift	50	0.305	0.77	0.70	11.21	25.86	0.0	0.0	0.09	0.29	**0.69**	0.59
		0.53	0.83	0.78	11.78	21.53	0.0	0.0	0.56	0.51	**0.73**	0.72
		0.755	0.92	0.71	6.95	28.29	0.0	0.0	0.69	0.60	**0.86**	0.85
		0.98	0.97	0.71	6.72	30.07	0.0	0.0	0.85	0.59	**0.89**	0.82
		1.205	0.95	0.69	6.51	28.30	0.0	0.0	0.87	0.73	**0.93**	0.86
		1.43	1.00	0.80	8.58	23.14	0.0	0.0	0.90	0.81	**0.96**	0.84
Systematic	30	0.08	0.45	0.53	7.74	6.70	0.0	0.0	0.03	0.05	0.32	**0.40**
		0.155	0.52	0.47	3.75	10.65	0.0	0.0	0.01	0.05	0.29	**0.38**
		0.23	0.57	0.45	3.26	9.69	0.0	0.0	0.01	0.09	0.27	**0.44**
		0.305	0.63	0.47	3.72	37.19	0.0	0.1	0.01	0.10	0.28	**0.41**
		0.38	0.66	0.41	5.32	34.95	0.0	0.1	0.01	0.05	0.27	**0.35**
		0.455	0.76	0.45	5.11	33.61	0.0	0.0	0.01	0.10	0.28	**0.42**
Cyclic	50	0.605	0.78	0.51	18.77	21.65	0.0	0.0	0.17	0.02	**0.45**	0.26
		0.68	0.80	0.47	12.67	38.50	0.0	0.0	0.16	0.02	**0.48**	0.30
		0.755	0.81	0.54	12.88	25.65	0.0	0.0	0.37	0.02	**0.53**	0.29
		0.83	0.81	0.58	11.66	31.19	0.0	0.0	0.09	0.06	**0.56**	0.31
		0.905	0.81	0.72	11.52	25.38	0.0	0.0	0.33	0.22	**0.63**	0.29
		0.98	0.84	0.61	11.54	34.43	0.0	0.0	0.24	0.04	**0.64**	0.40
Stratification	80	0.605	0.65	0.24	8.51	56.77	0.0	0.1	0.16	0.02	**0.36**	0.30
		0.68	0.70	0.34	13.55	65.15	0.0	0.2	0.05	0.13	0.37	**0.40**
		0.755	0.73	0.00	5.56	53.43	0.0	0.1	0.20	0.04	**0.45**	0.35
		0.83	0.85	0.34	10.68	59.46	0.0	0.0	0.33	0.07	**0.47**	0.35
		0.905	0.79	0.58	11.82	55.61	0.0	0.0	0.35	0.19	0.30	**0.32**
		0.98	0.93	0.63	8.21	52.23	0.0	0.1	0.16	0.16	**0.51**	0.30

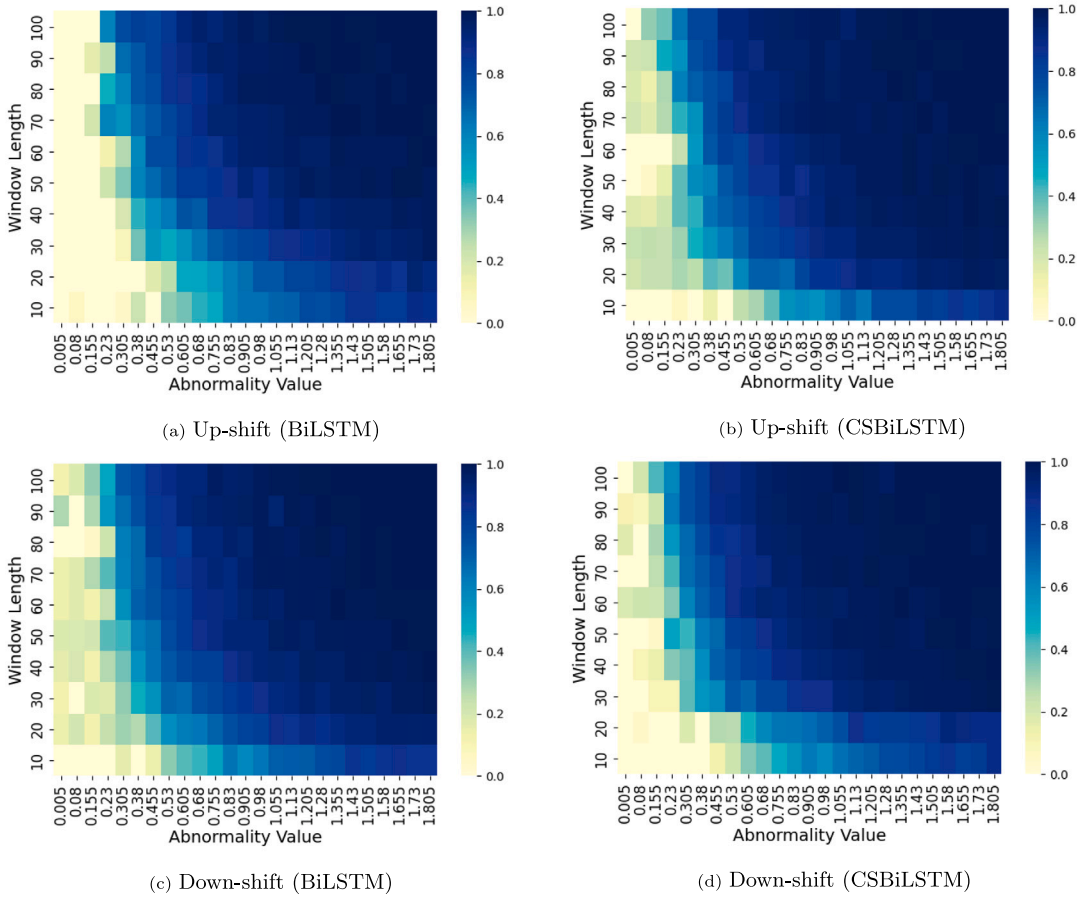


Fig. 14. Obtained boundary for inseparable, partially separable, and separable classification for BiLSTM and CSBiLSTM methods in up-shift and down-shift patterns.

Table 12
Hyperparameters for the viable cell density dataset. N_l and δ_l denote the number of neurons and dropout rate in layer l.

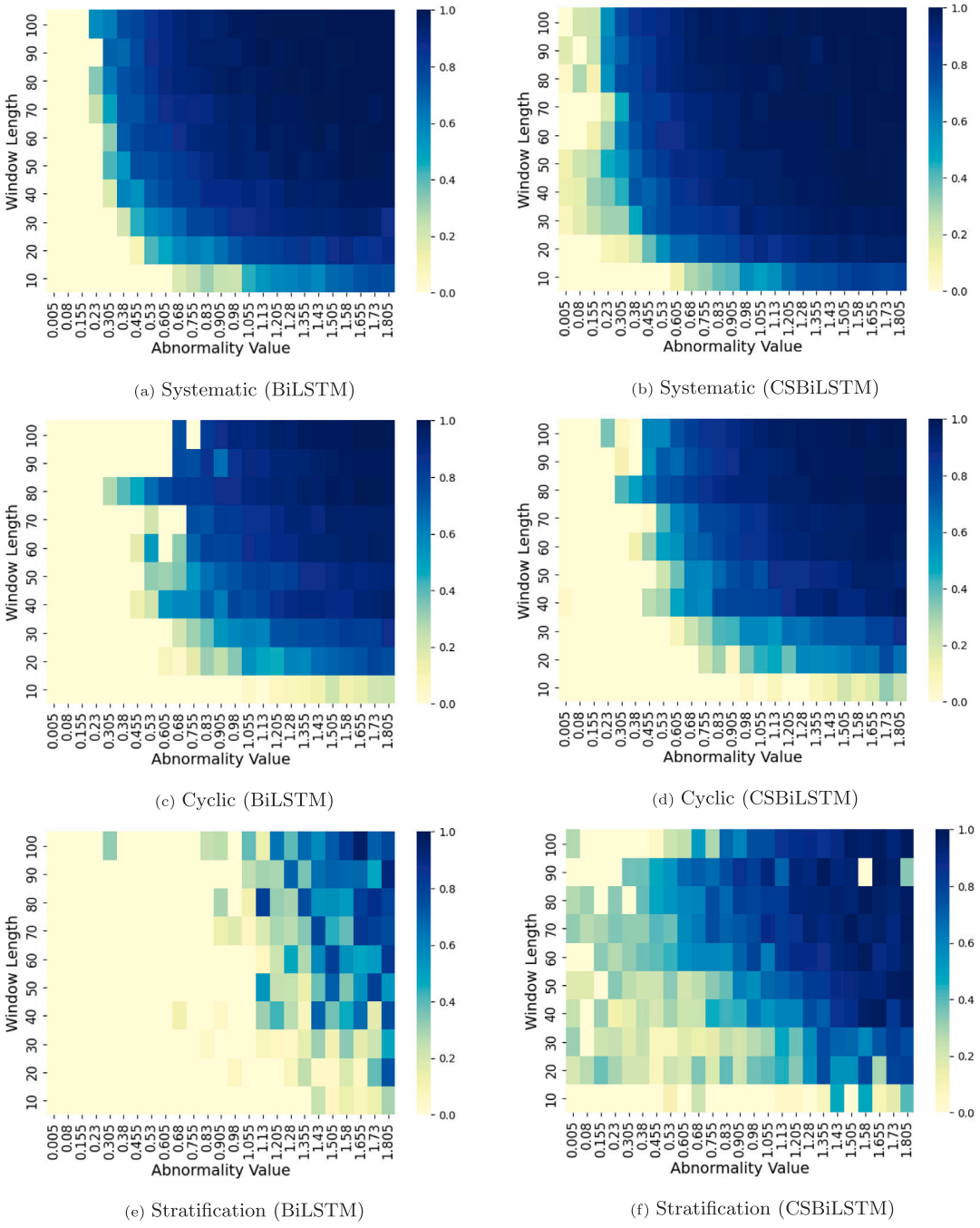
(ρ, κ, MRW, λ)	Layer (l)	(N_l, δ_l)
(0.0067, 14.4, E, 4)	1	(9, 0.3)
	2	(8, 0.4)
	3	(4, 0.5)

3.5. Early detection analysis

In this section, we discuss our approach to evaluate and analyze the early detection strengths of the four distinct methods introduced earlier. We randomly selected representative problems for each pattern with a range of window lengths $T \in \{30, 50, 80\}$ and, 25 different abnormality values, covering a wide spectrum from inseparable instances to easily separable ones. For each pattern, we generate two datasets, each comprising 3000 samples, using MRDS and AUDS methods. We implement four classifiers of BiLSTM-AUDS, CSBiLSTM-AUDS, BiLSTM-MRDS, and CSBiLSTM-MRDS. Furthermore, we study the sensitivity of these methods with respect to various class imbalance ratio values. We vary the class imbalance ratio from 0.5, 0.75, 0.9, and 0.95, which represents a spectrum from balanced scenarios to extreme class imbalance.

For this experiment, we apply a dynamic rolling window technique and set the maximum detection horizon at 120 steps. Our test set has a total of 300 abnormal samples. After normalizing and applying predictions using the trained classifiers, we report the earliest time at which a rolled test instance is successfully detected. This value represents the time required to detect an abnormality under a specific imbalanced ratio, window length, and abnormality value. To better measure the speed at which each method detects abnormalities, we compute the ARLIDX metric using a set of 300 test samples. Fig. 17 illustrates this metric for the up-shift pattern with various abnormal parameters and training approaches. The figure demonstrates that MRDS consistently results in lower ARLIDX values, indicating a faster detection of abnormalities. This trend remains consistent across different window lengths and imbalanced ratios, regardless of the choice of training classifiers. Furthermore, the ARLIDX metric shows a similar pattern for imbalanced datasets as the window length increases from 30 to 80 across different approaches. For balanced datasets, MRDS-based BiLSTM and CSBiLSTM results are similar. With shorter window lengths, the ARLIDX tends to be smaller because abnormal signals are more likely to be detected earlier and more rapidly.

Moreover, our experiments demonstrate that lower abnormality values lead to more challenging detection problems, while higher values indicate relatively easier problems. This observation is evident from our results, which are presented in Section 3.4 and illustrated in Fig. 18. As shown in Fig. 18, it is apparent that the CSBiLSTM using MRDS method outperforms other combined training and data generation methods in terms of robustness, characterized by its lower variability in ARLIDX as parameter pattern changes. On the other hand, in some cases, particularly in highly imbalanced datasets, the trained classifier may be biased toward the normal class, leading to detection failures throughout the entire rolling horizon attempts. Tables 8 and 9 display the USP index, which provides a summary of the percentage of these failures among all 300 test data samples for various problems and methods. The results indicate that CSBiLSTM with MRDS outperforms the other methods, consistently achieving lower USP values across nearly all



(a) Systematic (BiLSTM)

(b) Systematic (CSBiLSTM)

(c) Cyclic (BiLSTM)

(d) Cyclic (CSBiLSTM)

(e) Stratification (BiLSTM)

(f) Stratification (CSBiLSTM)

Fig. 15. Obtained boundary for inseparable, partially separable, and separable classification problems for systematic, cyclic, and stratification patterns.

Table 13
Stratified 4-fold cross-validation CSBiLSTM results in terms of G-mean for the viable cell density dataset.

Fold ID	G-mean over rolling windows					
	1^{st} Roll	2^{nd} Roll	3^{rd} Roll	4^{th} Roll	5^{th} Roll	6^{th} Roll
Fold 1	0.89	0.89	0.89	1.00	1.00	1.00
Fold 2	1.00	1.00	1.00	1.00	1.00	1.00
Fold 3	0.61	0.61	1.00	1.00	1.00	1.00
Fold 4	1.00	1.00	1.00	1.00	1.00	1.00
Average	0.88	0.88	0.97	1.00	1.00	1.00

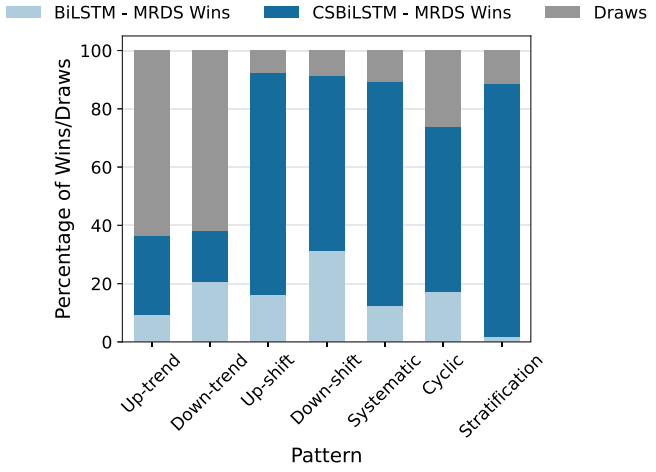


Fig. 16. Wins and draws in instance-to-instance G-Mean comparison of BiLSTM and CSBiLSTM.

Table 14
Stratified 4-fold cross-validation CSBiLSTM results in terms of early detection metrics for the viable cell density dataset.

Fold ID	ARLIDX	USP	TASRID
Fold 1	1.6	0	0.9
Fold 2	1	0	1
Fold 3	1.5	0	0.92
Fold 4	1	0	1
Average	1.28	0	0.95

combinations of abnormality values, imbalanced ratios, and window lengths.

Figs. 19 illustrate the TASRID results for the up-shift pattern with class imbalance ratios of 50%, 75%, 90%, and 95%, and window lengths of 30, 50, and 80. One can observe that the MRDS methods outperform the AUDS methods for early detection in most cases. Furthermore, we have observed that the cost-sensitive methods exhibit superior performance in comparison to their cost-insensitive counterparts. This trend highlights the efficacy of the MRDS approach and the advantages of cost-sensitive learning techniques in obtaining a more stable classifier. As the class imbalance ratio increases, the TASRID slightly decreases across all classifiers. As we observe variations in imbalance ratios across different window lengths, the classifiers show reduced consistency in generating true predictions. The classifier's consistency in true detections increases with higher TASRID values. For balanced data, the TASRID value of CS-BiLSTM with AUDS is slightly lower compared to the BiLSTM with AUDS method. Furthermore, we have obtained similar results for other patterns.

In addition, we assess TASRID values across various abnormality values, as shown in Fig. 20. A consistent trend emerges as we observe that decreasing abnormality values make problem instances more challenging, resulting in higher false alarm rates across classifiers and lower TASRID values. This observation is consistent across various window lengths and imbalance ratios. Lower abnormality values make early detection more complex. It is worth mentioning that with lower abnormality values, the CSBiLSTM with MRDS method consistently reveals higher TASRID values compared to the other three methods, which highlights its capability to provide reliable and stable predictions. Therefore, this method excels in reliable early detection.

3.6. Comparative analysis with Ünlü (2021a)'s CSBiLSTM

Next, we compare our CSBiLSTM model with the cost-sensitive BiLSTM proposed by Ünlü (2021a), based on G-mean, ARLIDX, USP, and TASRID to measure the performance accuracy as well as early and stable abnormality detection. Furthermore, the unified score is reported for the sake of simple comparison. Tables 10 and 11 detail our results.

Our CSBiLSTM model demonstrates superior performance compared to the Ünlü's BiLSTM model across various metrics for most patterns, particularly in terms of G-mean and the unified score. Furthermore, our method achieves faster and more consistent detection of abnormalities, as indicated by lower ARLIDX and higher TASRID values, ensuring prompt identification of anomalies.

3.7. Real-world applications

For real-world applications, the process of building our model starts with data collection. This is done through identifying key process variables followed by collecting and storing relevant physical parameters (e.g., temperature and viable cell density) through sensors and machine controllers (e.g., within a supervisory control and data acquisition (SCADA) system or more modern data acquisition systems) or through manual data entries. For Control charts can then be used to identify when the process deviates from its expected range of variability by plotting the data points and determining statistical limits (control limits) to detect out-of-control samples. The model is then constructed using a diverse and representative dataset that includes a wide range of features related to the process being modeled (e.g., collected parameters), along with samples from both normal operating condition and anomalies (or failures) must be used as the input. Careful tuning of the model parameters is done through validation (e.g., cross-validation method). The constructed model can then identify an abnormal sample by predicting its label. Notably, the model is essentially based on historical data: frequent update in the input data must be performed while ensuring that the data always contains enough samples from both normal and abnormal patterns.

In this section, we study the application of our CSBiLSTM and the proposed TASRID and USP as well as ARLIDX measures using two real-world datasets collected from the pharmaceutical and wafer manufacturing industries. Specifically, we explain the steps for data collection and pre-processing, illustrate the adaptation of the model for these specific industry datasets, and validate its capability in detecting anomalies using our proposed metrics.

3.7.1. Pharmaceutical manufacturing industry

We implemented our proposed algorithms in a pharmaceutical manufacturing industry, where a quick anomaly detection is critical for ensuring product safety and efficacy. Current biomanufacturing procedures are still guided by human expert knowledge and, therefore, struggle to effectively handle the complex bioprocesses (Park, Park,

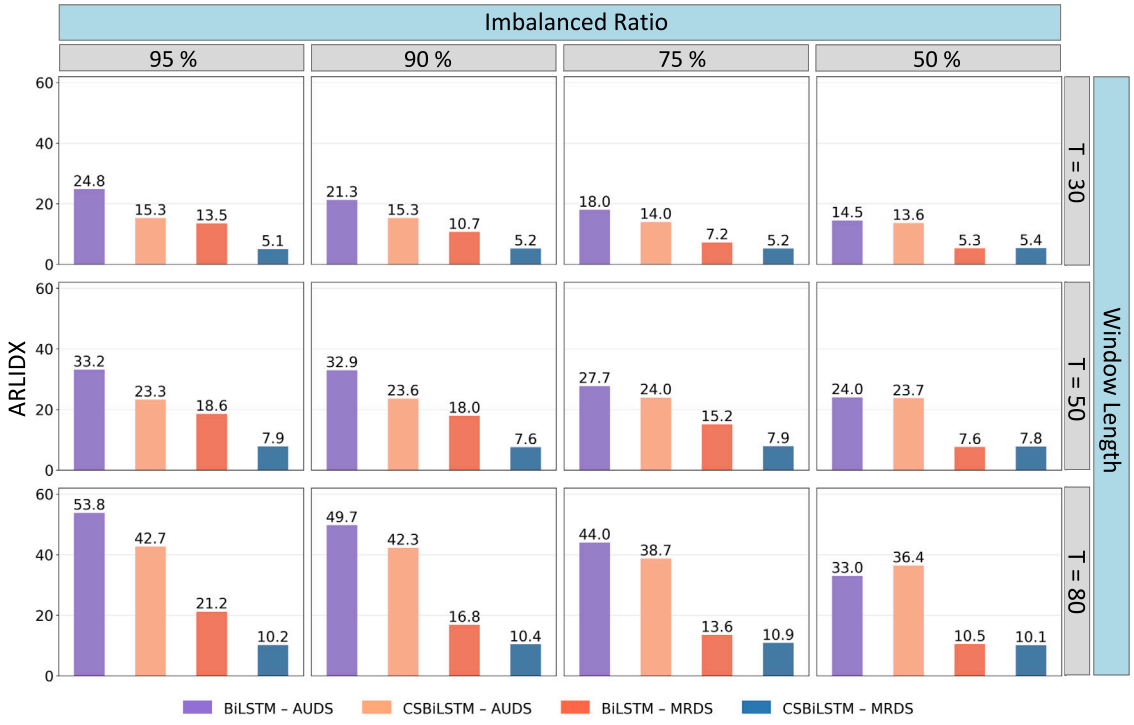


Fig. 17. ARLIDX comparison of four methods for the up-shift pattern.

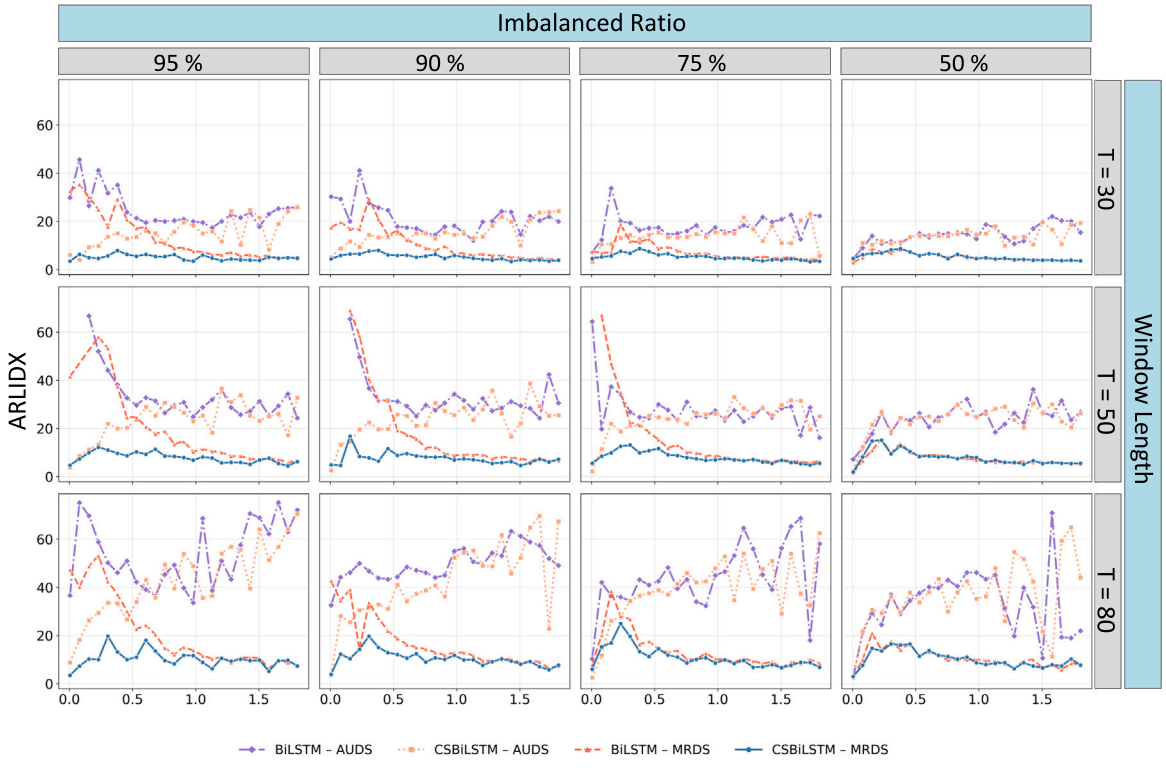


Fig. 18. ARLIDX results for the up-shift pattern using four methods.

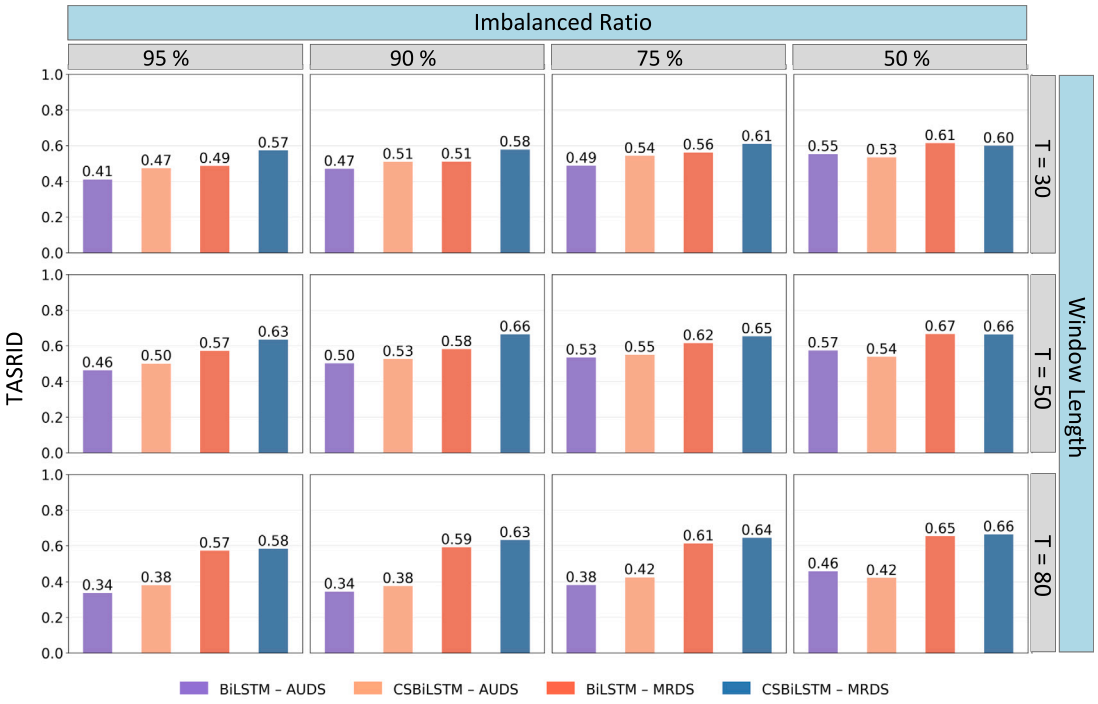


Fig. 19. TASRID analysis for four methods for the up-shift pattern.

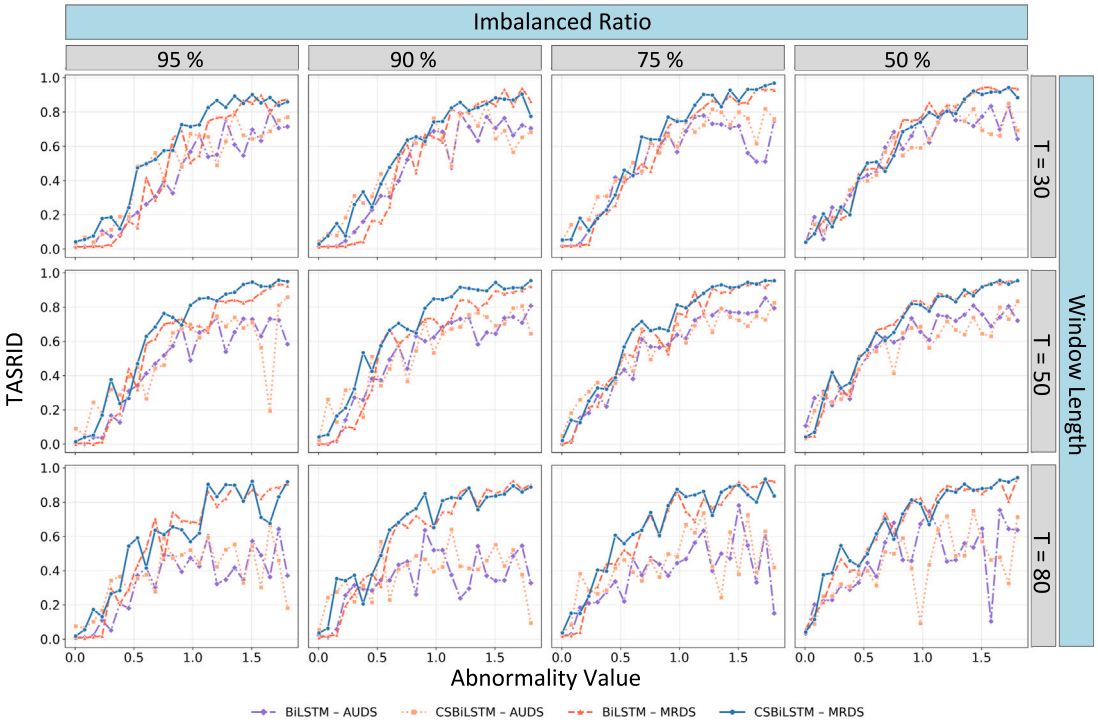


Fig. 20. TASRID results of four methods for up-shift pattern based on abnormality value and class imbalance ratio.

Choi, Hong, & Lee, 2021). By leveraging a network of sensors attached to the bioreactors, it is possible to continuously monitor the quality of bio-pharmaceutical products. These sensors focus on several critical variables, including viable cell density, pH, and cell viability, to ensure optimal product quality.

In this study, we have employed viable cell density as a key indicator for monitoring bioprocesses in mammalian cell cultures due to its strong correlation with the quality and efficacy of the biomanufactured products. This variable measures the concentration of living cells within a culture, specifically focusing on Chinese hamster ovary (CHO) cells, which have been a cornerstone in biomanufacturing. The dataset on viable cell density includes 24 time series, each representing data collected throughout a 15-day production cycle. Fig. 21 shows the pattern indicative of down-trends starting day 8, which signal the production of low-quality or defective products. In contrast, time series data that show stable trends are labeled as normal, consistent with the expected desirable product quality. Fig. 21 illustrates this distinction, where normal samples are represented in blue, and abnormal samples, characterized by downtrend patterns, are marked in red. After this labeling step, we observe the class imbalance ratio of 75%. Using transfer learning and the observed down-trend pattern, we employ a similar optimized architecture for the down-trend pattern, as detailed in Table 6, and further described in Table 12.

In the implementation of our CSBiLSTM model for anomaly detection using data from a pharmaceutical company, we utilize stratified 4-fold cross-validation alongside the RW technique to handle the small dataset. The data on viable cell density is divided into four segments, with each segment alternately serving as the test set in repeated trials. Prior to the start of training, the data is normalization. It is crucial for practitioners to retain the fitted scaler, ensuring the data is consistently normalized when applying the RW technique to test data. Fig. 21 provides a schematic overview of our approach.

The hyperparamters selected for this dataset are summarized in Table 12. The G-mean values obtained for each cross-validated test set are summarized in Table 13. As one can observe, the G-mean improves over the rolling horizon, demonstrating the effectiveness of our approach in detecting abnormalities. Furthermore, we calculated the early detection metrics for this dataset, as demonstrated in Table 14. The last row of the table represents the average across different folds when they were utilized as the test sets. Our findings demonstrate the ability to detect accurately abnormal manufacturing processes within a day or two, as indicated by an ARLIDX value of 1.28. The lack of undetected sequences and the higher TASRID values show the reliability of the trained classifier in early detecting abnormalities.

3.7.2. Wafer manufacturing industry

Similarly, we evaluate our proposed method through its application in a real-life scenario within the wafer manufacturing industry. The wafer dataset, introduced by Dau et al. (2019), consists of two distinct sets: a training set containing 1000 samples and a test set containing 6164 samples. It is worth noting that this dataset is imbalanced, where the majority class represents approximately 90% of both the training and test sets. Each time series in this dataset has a length of 152. We allocated 20% of the training data as a validation set. We have applied the RW technique to assess our proposed TASRID index on the various problem instances, considering different combinations of window sizes and stride levels. We utilize the stride level to account for the time when new data is introduced into the system. Furthermore, to maintain comparability across various window sizes and strides, we calculated the ARLIDX indicator for each test data sample using the formula $T + F \times S$, where T represents the window length, F represents the rolling index at which the test data sample is classified correctly for the first time, and S denotes the stride value. The results of these experiments are illustrated in Table 16. We have observed the down-shift pattern in wafer data. By leveraging transfer learning, we can take advantage of an already optimized architecture tailored for this specific pattern, as detailed in Table 6, and further described in Table 15.

We observe that selecting the appropriate window length for monitoring the production process is critical. For instance, when the stride value is set to 1, a window length of 10 or 30 will result in lower USP and ARLIDX values. However, with these window lengths, the trained classifier shows lower TASRID scores, signifying potential instability in early detection. Further, we have noticed a substantial improvement in the TASRID index when the window length is set to or exceeds 60. This observation, along with the lower USP and ARLIDX values, supports a recommended window length of greater than 60. We have observed a similar trend for other stride values.

4. Conclusion and future directions

In this paper, we developed a predictive model based on BiLSTM neural networks for control chart pattern recognition, a critical component of advanced fault detection in digital twins for manufacturing systems. We particularly employed the cost-sensitive focal loss function to address the severe class imbalance, which remains an inherent issue in control chart pattern recognition datasets. We further developed a bi-objective early stopping technique that optimally balances loss minimization and G-mean maximization during model training. Additionally, we introduced a Bayesian optimization-based approach to fine-tune hyperparameters for each abnormal pattern.

Through the novel data simulation scheme generating abnormal patterns in real-time, we developed an innovative adaptive weighting strategy. This strategy effectively trains BiLSTM classifiers to be more sensitive to abnormal data points with either higher or lower abnormal signals. This sensitivity contributes to their early detection. In addition, it also takes into account the class imbalance within mini-batches. Through an extensive set of experiments, we demonstrated the superior performance of our proposed methods in swiftly and accurately recognizing abnormal patterns. We further validated our method on real-world datasets from wafer manufacturing and bio-manufacturing industries.

While our work demonstrated the benefits of using CSBiLSTM to address class imbalance and facilitate early detection of control chart patterns within the DT paradigm, several promising avenues for future research exist. We would like to extend the deployment of our model to real-world applications and fine-tune its ability to handle dynamic, real-time data. We demonstrated that the specific onset of abnormality significantly impacts the classifier's early detection performance. Future research may explore the development of more advanced BiLSTM models to learn complex abnormal patterns, especially in mixed-signal pattern recognition problems with varying signal ratios. Another future research direction is the pursuit of optimal models based on combined classification metrics and early detection measures, further advancing CCPR methodologies. In summary, this research opens new avenues for the application of DT-based models in control chart pattern recognition, promising significant advancements in fault detection systems through future investigations in these directions.

Furthermore, we employed the CSBiLSTM model for anomaly detection in an offline manner for both pharmaceutical and wafer manufacturing industries. This approach relies solely on historical data and does not adapt to new information after its initial deployment. While it is effective at identifying initial anomalies, this approach might not accurately identify when processes return to normal operation after corrective measures. To address this limitation, we propose online learning as a promising avenue for future investigation. Online learning would allow for the continuous update of the model with new data, facilitating its refinement through subsequent rounds of post-repair testing and adjustments. This could significantly enhance the model's adaptability and effectiveness in dynamic manufacturing environments.

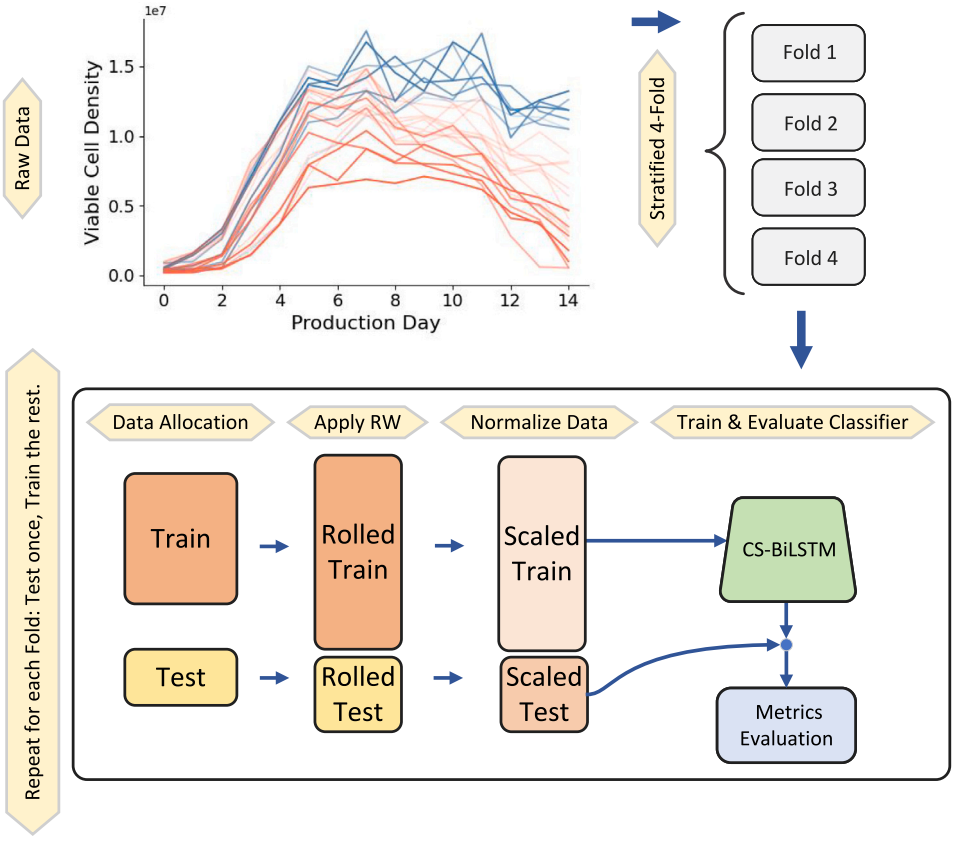


Fig. 21. The overview of our approach for the viable cell density dataset. In the first figure, the normal pattern is shown by blue color, and the abnormal pattern is shown by red color.

Table 15
Hyperparameters for the wafer data. N_l and δ_l denote the number of neurons and dropout rate in layer l.

$(\rho, \kappa, \text{MRW}, \lambda)$	T	(N_l, δ_l) for layers $l = 1, 2, \ldots, 5$				
		1	2	3	4	5
(0.0031, 12, D, 1)	10	(10, 0.3)	(8, 0.5)	(7, 0.2)	(6, 0.6)	(6, 0.3)
	20	(17, 0.6)	(16, 0)	(11, 0.6)	(8, 0.5)	(5, 0.3)
	30	(29, 0.3)	(24, 0.5)	(16, 0.4)	(14, 0.6)	(8, 0.3)
	40	(35, 0.2)	(27, 0.4)	(23, 0.4)	(19, 0.5)	(6, 0.4)
	50	(46, 0.4)	(39, 0.4)	(25, 0.2)	(23, 0.5)	(9, 0.3)
	60	(57, 0.1)	(46, 0.5)	(31, 0.4)	(24, 0.2)	(12, 0.4)
	70	(60, 0.3)	(50, 0.2)	(33, 0.2)	(30, 0.5)	(6, 0.5)
	80	(72, 0.2)	(41, 0.1)	(50, 0.4)	(31, 0.3)	(16, 0.1)
	90	(76, 0.2)	(61, 0.4)	(47, 0.4)	(34, 0.3)	(8, 0.4)
	100	(92, 0.2)	(75, 0.4)	(59, 0.4)	(31, 0.3)	(20, 0.2)

Table 16
Early detection metrics for wafer dataset for different window lengths and stride values for wafer data.

Metric	Stride	T									
		10	20	30	40	50	60	70	80	90	100
ARLIDX	1	18.06	30.72	32.77	40.02	50.50	60.03	70.03	80.02	90.04	100.01
	5	18.22	24.60	31.07	40.09	50.05	60.08	70.01	80.02	90.01	100.01
	10	21.95	22.80	31.95	40.16	50.38	60.05	70.17	80.01	90.03	100.02
USP	1	0.00%	0.05%	0.00%	0.04%	0.00%	0.00%	0.00%	0.00%	0.00%	0.02%
	5	0.02%	0.09%	0.00%	0.00%	0.02%	0.05%	0.00%	0.02%	0.00%	0.02%
	10	0.00%	0.02%	0.02%	0.00%	0.00%	0.04%	0.07%	0.00%	0.00%	0.02%
TASRID	1	0.113	0.126	0.201	0.607	0.391	0.905	0.837	0.915	0.974	0.983
	5	0.098	0.256	0.375	0.571	0.575	0.805	0.799	0.914	0.975	0.989
	10	0.173	0.339	0.393	0.584	0.694	0.828	0.818	0.977	0.984	0.988

CRediT authorship contribution statement

Mohammad Derakhshi: Conceptualization, Modeling, Methodology, Validation, Software, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Talayeh Razzaghi:** Proposal & original idea, Conceptualization, Modeling, Methodology, Validation, Formal analysis, Investigation, Data curation, Visualization, Writing – original draft, Writing – review & editing, Supervision, Project administration, Funding acquisition.

Declaration of competing interest

None of the authors of this work have an interest to declare.

Data availability

The data that has been used is confidential. Our source code and documentation are available at: https://github.com/Deraxsi/Imbalance-Aware-LSTM-CCP-Early-Detection.git.

Acknowledgments

The authors would like to thank Cytovance Biologics for their generous contribution of biomanufacturing datasets and their close collaboration during the research project. The authors are further grateful for the support by "OKC ED Fd: OU Biotech Core Facility", Oklahoma City Chamber of Commerce, U.S. Economic Development Administration (Award #: 08-79-05677). All authors have read and agreed to the published version of the manuscript.

References

Aljubran, M., Ramasamy, J., Albassam, M., & Magana-Mora, A. (2021). Deep learning and time-series analysis for the early detection of lost circulation incidents during drilling operations. *IEEE Access*, *9*, 76833–76846.

Bao, W., Yue, J., & Rao, Y. (2017). A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS One*, *12*(7), Article e0180944.

Barricelli, B. R., Casiraghi, E., & Fogli, D. (2019). A survey on digital twin: Definitions, characteristics, applications, and design implications. *IEEE Access*, *7*, 167653–167671.

Bergstra, J., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. *Advances in Neural Information Processing Systems*, *24*.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., et al. (2015). The UCR time series classification archive. www.cs.ucr.edu/~eamonn/time_series_data/.

Chen, T., Xu, R., He, Y., & Wang, X. (2017). Improving sentiment analysis via sentence type classification using BiLSTM-CRF and CNN. *Expert Systems with Applications*, *72*, 221–230.

Chollet, F., et al. (2015). Keras. GitHub. URL https://github.com/fchollet/keras.

Company, W. E. (1956). *Statistical quality control handbook*. Western Electric Company.

Dau, H. A., Bagnall, A., Kamgar, K., Yeh, C.-C. M., Zhu, Y., Gharghabi, S., et al. (2019). The UCR time series archive. *IEEE/CAA Journal of Automatica Sinica*, *6*(6), 1293–1305.

De la Torre Gutiérrez, H., & Pham, D. T. (2018). Identification of patterns in control charts for processes with statistically correlated noise. *International Journal of Production Research*, *56*(4), 1504–1520.

Dokuz, Y., & Tufekci, Z. (2021). Mini-batch sample selection strategies for deep learning based speech recognition. *Applied Acoustics*, *171*, Article 107573.

Elkan, C. (2001). The foundations of cost-sensitive learning. *Vol. 17*, In *International Joint Conference on Artificial Intelligence* (1), (pp. 973–978). Lawrence Erlbaum Associates Ltd.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*(2), 179–211.

Fernández, A., Garcia, S., Herrera, F., & Chawla, N. V. (2018). SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary. *Journal of Artificial Intelligence Research*, *61*, 863–905.

Fuqua, D., & Razzaghi, T. (2020). A cost-sensitive convolution neural network learning for control chart pattern recognition. *Expert Systems with Applications*, *150*, Article 113275.

García, E., Peñabaena-Niebles, R., Jubiz-Diaz, M., & Perez-Tafur, A. (2022). Concurrent control chart pattern recognition: A systematic review. *Mathematics*, *10*(6), 934.

Graves, A., Liwicki, M., Fernández, S., Bertolami, R., Bunke, H., & Schmidhuber, J. (2008). A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(5), 855–868.

Graves, A., Mohamed, A.-r., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing* (pp. 6645–6649). http://dx.doi.org/10.1109/ICASSP.2013.6638947.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, *18*(5–6), 602–610.

Hachicha, W., & Ghorbel, A. (2012). A survey of control-chart pattern-recognition literature (1991–2010) based on a new conceptual classification scheme. *Computers & Industrial Engineering*, *63*(1), 204–222.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, *79*(8), 2554–2558.

Hutter, F., Hoos, H. H., & Leyton-Brown, K. (2011). Sequential model-based optimization for general algorithm configuration. In *Learning and Intelligent Optimization: 5th International Conference, LION 5, Rome, Italy, January 17-21, 2011. selected papers 5* (pp. 507–523). Springer.

Hwarng, H. B., & Hubele, N. F. (1993a). Back-propagation pattern recognizers for X control charts: Methodology and performance. *Computers & Industrial Engineering*, *24*(2), 219–235.

Hwarng, H. B., & Hubele, N. F. (1993b). X control chart pattern identification through efficient off-line neural network training. *IIE Transactions*, *25*(3), 27–40.

Jordan, M. I. (1997). Serial order: A parallel distributed processing approach. In *Advances in Psychology*: *vol. 121*, (pp. 471–495). Elsevier.

Kizilirmak, F., & Yanikoglu, B. (2023). CNN-BiLSTM model for English handwriting recognition: Comprehensive evaluation on the IAM dataset. arXiv preprint arXiv:2307.00664.

Knoth, S. (2006). The art of evaluating monitoring schemes–how to measure the performance of control charts? *Vol. 8*, In *Frontiers in statistical quality control* (pp. 74–99). Springer.

Kuo, T., & Mital, A. (1993). Quality control expert systems: A review of pertinent literature. *Journal of Intelligent Manufacturing*, *4*, 245–257.

Li, B., Liu, Y., & Wang, X. (2019). Gradient harmonized single-stage detector. *Vol. 33*, In *Proceedings of the AAAI Conference on Artificial Intelligence* (01), (pp. 8577–8584).

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 2980–2988).

Liu, C., Chen, K., Jin, S., & Qu, Y. (2019). Variation pattern recognition of the BIW OCMM online measurement data based on LSTM NN. *IEEE Access*, *7*, 69007–69014.

Lu, Y., Cheung, Y.-m., & Tang, Y. Y. (2017). Dynamic weighted majority for incremental learning of imbalanced data streams with concept drift. In *IJCAI* (pp. 2393–2399).

Lu, Z., Wang, M., & Dai, W. (2020). A condition monitoring approach for machining process based on control chart pattern recognition with dynamically-sized observation windows. *Computers & Industrial Engineering*, *142*, Article 106360.

Maged, A., & Xie, M. (2023). Recognition of abnormal patterns in industrial processes with variable window size via convolutional neural networks and AdaBoost. *Journal of Intelligent Manufacturing*, *34*(4), 1941–1963.

Malialis, K., Panayiotou, C. G., & Polycarpou, M. M. (2020). Online learning with adaptive rebalancing in nonstationary environments. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(10), 4445–4459.

Marchetti, F., Guastavino, S., Piana, M., & Campi, C. (2022). Score-oriented loss (sol) functions. *Pattern Recognition*, *132*, Article 108913.

Martín, A., Ashish, A., Paul, B., Eugene, B., Zhifeng, C., Craig, C., et al. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. https://www.tensorflow.org.

Miao, Z., & Yang, M. (2019). Control chart pattern recognition based on convolution neural network. In *Smart Innovations in Communication and Computational Sciences: Proceedings of ICSICCs 2017, Vol. 2* (pp. 97–104). Springer.

Pacella, M., & Semeraro, Q. (2007). Using recurrent neural networks to detect changes in autocorrelated processes for quality monitoring. *Computers & Industrial Engineering*, *52*(4), 502–520.

Park, S.-Y., Park, C.-H., Choi, D.-H., Hong, J. K., & Lee, D.-Y. (2021). Bioprocess digital twins of mammalian cell culture for advanced biomanufacturing. *Current Opinion in Chemical Engineering*, *33*, Article 100702.

Peng, D., Gu, T., Hu, X., & Liu, C. (2021). Addressing the multi-label imbalance for neural networks: An approach based on stratified mini-batches. *Neurocomputing*, *435*, 91–102.

Pham, D., & Oztemel, E. (1992). XPC: An on-line expert system for statistical process control. *The International Journal of Production Research*, *30*(12), 2857–2872.

Shao, Y. E., & Chiu, C.-C. (2016). Applying emerging soft computing approaches to control chart pattern recognition for an SPC–EPC process. *Neurocomputing*, *201*, 19–28.

Shimizu, R., Asako, K., Ojima, H., Morinaga, S., Hamada, M., & Kuroda, T. (2018). Balanced mini-batch training for imbalanced image data classification with neural network. In *2018 First International Conference on Artificial Intelligence for Industries* (pp. 27–30). IEEE.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. *Advances in Neural Information Processing Systems*, *25*.

Swift, J. A., & Mize, J. H. (1995). Out-of-control pattern recognition and analysis for quality control charts using lisp-based systems. *Computers & Industrial Engineering*, *28*(1), 81–91.

Tran, P. H., Ahmadi Nadi, A., Nguyen, T. H., Tran, K. D., & Tran, K. P. (2022). Application of machine learning in statistical process control charts: A survey and perspective. In *Control Charts and Machine Learning for Anomaly Detection in Manufacturing* (pp. 7–42). Springer.

Ünlü, R. (2021a). Cost-oriented LSTM methods for possible expansion of control charting signals. *Computers & Industrial Engineering*, *154*, Article 107163.

Ünlü, R. (2021b). A robust data simulation technique to improve early detection performance of a classifier in control chart pattern recognition systems. *Information Sciences*, *548*, 18–36.

Wang, Q., Ma, Y., Zhao, K., & Tian, Y. (2020). A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, 1–26.

Wang, S., Minku, L. L., & Yao, X. (2018). A systematic study of online class imbalance learning with concept drift. *IEEE Transactions on Neural Networks and Learning Systems*, *29*(10), 4802–4821.

Wang, N., Zhao, X., Jiang, Y., Gao, Y., & BNRist, K. (2018). Iterative metric learning for imbalance data classification. *2018*, In *IJCAI* (pp. 2805–2811).

Wei, M., Xu, Z., & Hu, J. (2021). Entity relationship extraction based on Bi-LSTM and attention mechanism. In *2021 2nd International Conference on Artificial Intelligence and Information Systems* (pp. 1–5).

Wu, C., Liu, F., & Zhu, B. (2015). Control chart pattern recognition using an integrated model based on binary-tree support vector machine. *International Journal of Production Research*, *53*(7), 2026–2040.

Xanthopoulos, P., & Razzaghi, T. (2014). A weighted support vector machine method for control chart pattern recognition. *Computers & Industrial Engineering*, *70*, 134–149.

Xu, P., Du, R., & Zhang, Z. (2019). Predicting pipeline leakage in petrochemical system through GAN and LSTM. *Knowledge-Based Systems*, *175*, 50–61.

Xue, L., Wu, H., Zheng, H., & He, Z. (2023). Control chart pattern recognition for imbalanced data based on multi-feature fusion using convolutional neural network. *Computers & Industrial Engineering*, *182*, Article 109410.

Yeung, M., Sala, E., Schönlieb, C.-B., & Rundo, L. (2022). Unified focal loss: Generalising dice and cross entropy-based losses to handle class imbalanced medical image segmentation. *Computerized Medical Imaging and Graphics*, *95*, Article 102026.

Yu, Y., Si, X., Hu, C., & Zhang, J. (2019). A review of recurrent neural networks: LSTM cells and network architectures. *Neural Computation*, *31*(7), 1235–1270.

Yu, J., Zheng, X., & Wang, S. (2019). A deep autoencoder feature learning method for process pattern recognition. *Journal of Process Control*, *79*, 1–15.

Zan, T., Liu, Z., Wang, H., Wang, M., & Gao, X. (2020). Control chart pattern recognition using the convolutional neural network. *Journal of Intelligent Manufacturing*, *31*, 703–716.

Zhang, W., Chen, Y., Yang, W., Wang, G., Xue, J.-H., & Liao, Q. (2020). Class-variant margin normalized softmax loss for deep face recognition. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(10), 4742–4747.

Zhu, Z., Dai, W., Hu, Y., & Li, J. (2020). Speech emotion recognition model based on Bi-GRU and focal loss. *Pattern Recognition Letters*, *140*, 358–365.