Digital Chemical Engineering 14 (2025) 100221

Contents lists available at ScienceDirect

Digital Chemical Engineering

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

Original article

Hyperbox Mixture Regression for process performance prediction in antibody production

Ali Nik-Khorasani [a], Thanh Tung Khuat [b,*], Bogdan Gabrys [b]

[a] Department of Electrical Engineering, Ferdowsi University of Mashhad, Mashhad, Iran
[b] Complex Adaptive Systems Laboratory, Data Science Institute, University of Technology Sydney, Sydney, Australia

ARTICLE INFO

Keywords:
Bioprocess performance prediction
Neuro-Fuzzy system
Hyperbox
Regression
Monoclonal antibodies

ABSTRACT

This paper addresses the challenges of predicting bioprocess performance, particularly in monoclonal antibody (mAb) production, where conventional statistical methods often fall short due to time-series data's complexity and high dimensionality. We propose a novel Hyperbox Mixture Regression (HMR) model that employs hyperbox-based input space partitioning to enhance predictive accuracy while managing uncertainty inherent in bioprocess data. The HMR model is designed to dynamically generate hyperboxes for input samples in a single-pass process, thereby improving learning speed and reducing computational complexity. Our experimental study utilizes a dataset that contains 106 bioreactors. This study evaluates the model's performance in predicting critical quality attributes in monoclonal antibody manufacturing over a 15-day cultivation period. The results demonstrate that the HMR model outperforms comparable approximators in accuracy and learning speed and maintains interpretability and robustness under uncertain conditions. These findings underscore the potential of HMR as a powerful tool for enhancing predictive analytics in bioprocessing applications.

1. Introduction

Regression models have found widespread application in various fields, including robot controllers (Li et al., 2020), motion prediction (Zhong et al., 2022), and time series forecasting (Lemke and Gabrys, 2010; Ruta et al., 2011; Zhang et al., 2024). Predicting bioprocess performance presents a complex multivariate time-series challenge that conventional statistical methods often struggle to address (Gangadharan et al., 2021). While numerous types of research have focused on data preprocessing techniques — such as imputation, visualization, and feature selection — choosing a suitable predictive model remains a critical hurdle (Khuat et al., 2024). This paper aims to tackle these challenges by developing a machine-learning model designed explicitly for bioprocess performance prediction.

The growing complexity of time-series data has led to an increasing reliance on machine learning (ML) techniques to overcome the limitations of traditional statistical methods. These conventional methods often struggle with the inherent correlations in time-series observations, resulting in potential inaccuracies in predictions (Gangadharan et al., 2019). In contrast, ML methods have gained popularity for extracting essential information from time-series data, providing more robust and accurate insights (Lim et al., 2023). This trend underscores a significant shift towards leveraging ML as a powerful tool for addressing the challenges inherent in time-series analysis.

In process engineering, mechanistic models based on physical, chemical, and biological principles are widely mentioned as robust tools. They are preferred by researchers because they provide valuable insights into the fundamental mechanisms of a process, enabling a deeper understanding of how different variables (e.g., temperature, pH, nutrient levels) influence system behavior. However, developing these models requires extensive domain knowledge of the underlying processes, such as the complex relationships between nutrients, metabolites, cells, and products, to accurately capture the growth or inhibition behaviors of cultured cells. This complexity often leads to overparameterized models with limited generalizability (Tsopanoglou and del Val, 2021). Additionally, the estimation of model parameters is significantly affected by noisy or inadequate experimental data. To ensure computational feasibility, mechanistic models rely on simplifying assumptions, which may fail to fully capture the complexity and heterogeneity of real-world bioprocesses. These models also lack the flexibility to adapt to dynamic or highly variable biological systems. The experimental cell culture datasets (Gangadharan et al., 2021) used in this study exemplify such real-world bioprocesses, characterized by heterogeneity in cell lines, cultivation conditions, and experimental setups, as well as the presence of noisy data. Consequently, mechanistic models are not well-suited for analyzing these datasets effectively. Recent studies have demonstrated the effectiveness of data-driven models

* Corresponding author.
E-mail addresses: al.nikkhorasani@mail.um.ac.ir (A. Nik-Khorasani), thanhtung.khuat@uts.edu.au (T.T. Khuat), bogdan.gabrys@uts.edu.au (B. Gabrys).

https://doi.org/10.1016/j.dche.2025.100221
Received 4 November 2024; Received in revised form 16 January 2025; Accepted 6 February 2025
Available online 15 February 2025

as an alternative to mechanistic models for predicting critical quality attributes (CQAs) and process outcomes, offering greater adaptability and accuracy in such complex systems. For instance, Khuat et al. (2024) highlighted the growing applications of ML in biopharmaceuticals, emphasizing its role in real-time monitoring and optimization of both upstream and downstream processes. By leveraging large datasets generated from production, ML models can identify patterns and relationships that are not easily discernible through conventional statistical methods. In addition, Waight et al. (2023) argued that identifying favorable biophysical properties is essential in the preclinical development of protein therapeutics, but predicting these properties remains challenging. They introduced an automated machine learning workflow that analyzes computationally derived features to build predictive models for key developability factors like hydrophobicity and polyspecificity in IgG molecules. More studies have been reviewed in Lim et al. (2023) and Khuat et al. (2024). This approach addresses some of the challenges in preclinical development, where predicting favorable biophysical properties is crucial yet difficult.

One of ML's key advantages is its ability to handle bioprocess data's high dimensionality and complexity. ML algorithms, such as random forests, support vector machines, and deep learning models, have been applied to predict CQAs and key performance indicators (KPIs) in monoclonal antibody (mAb) production. For instance, a recent study reports a deep learning-based 2D-convolutional neural network (2D-CNN) designed to predict various downstream processing attributes, including Protein A mAb elute concentration and aggregate percentages, from routinely collected process data (Alam et al., 2024). According to Alam et al. (2024), their model outperformed existing approaches, achieving a mean percentage deviation of less than 3% in experimental validation. In another study, Lai et al. (2022) employed machine learning to predict therapeutic antibody aggregation rates and viscosity at high concentrations ($150^{mg/ml}$), focusing on preclinical and clinical-stage antibodies. They employed a k-nearest neighbors regression model and achieved a high correlation for predicting aggregation rates using features derived from molecular dynamics simulations. Moreover, Schmitt et al. (2023) employed an Artificial Neural Network (ANN) to predict mAb viscosity. High concentrations of mAb solutions can increase viscosity, affecting protein purification and administration. They utilized an ANN and combined experimental factors and simulated data to predict and model the viscosity of mAbs.

Additionally, new research by Makowski et al. (2024) has shown the use of a transparent machine-learning model for predicting antibody (IgG1) variants with low viscosity based on the sequences of their variable (Fv) regions. This model identifies antibodies at risk for high viscosity with relatively high accuracy and enables the design of mutations that reduce antibody viscosity, confirmed experimentally. According to Makowski et al. (2024), their model demonstrated high accuracy and exhibited excellent generalization. These advancements underscore the growing role of ML and deep learning in enhancing the efficiency and quality of mAb production processes.

Regarding predicting mAb stability, the recent studies have focused on chemical modifications, such as methionine oxidation, that can impair antibody potency. For instance, a study developed a highly predictive in silico model for methionine oxidation by extracting features from mAb sequences, structures, and dynamics, utilizing random forests to identify crucial predictive features (Sankar et al., 2018). This work emphasized the potential for computational tools to complement experimental methods in therapeutic antibody discovery.

However, despite the potential benefits, challenges remain in the widespread adoption of ML in bioprocessing. Issues such as limited samples, high-dimensional data, data quality, model interpretability, and robust validation protocols must be addressed to ensure reliable application in industrial settings (Gangadharan et al., 2021, 2019; Khuat et al., 2024). In addition, the model should be capable of making inferences under uncertain conditions and, ideally, provide explanations for its predicted outcomes (Lim et al., 2023). While neuro-fuzzy regression models (Jang, 1993; de Campos Souza, 2020) effectively manage uncertainty, they face challenges in the high-dimensional data space (Pramod et al., 2019). Conversely, regression models such as Radial basis functions (RBF) and ANN, which excel in handling high-dimensional data spaces (Sung, 2004), struggle to address the inherent uncertainty in the problem. In addition, traditional machine learning models such as Multi-Layer Perceptrons (MLP) and Adaptive Neuro-Fuzzy Inference Systems (ANFIS) require multiple views of a sample and often struggle to learn from rare samples in imbalanced datasets. It motivates us to develop a neuro-fuzzy system that handles uncertainty within high-dimensional data spaces for bioprocess performance prediction.

RBF is a regression and classification model that can be employed when the relationship between variables is unknown. Key advantages of RBF models are guaranteed learning algorithm through linear least squares optimization and efficiency in dealing with high dimensional data (Walczak and Massart, 1996). This model consists of an unsupervised clustering framework that helps partition the input feature space, then estimate the target signal using least squares optimization (Tagliaferri et al., 2001; Walczak and Massart, 1996). However, a significant challenge for RBF networks is determining the optimal number and distribution of nodes in the hidden layer (Walczak and Massart, 1996).

Another class of powerful neuro-fuzzy machine learning models, which is of particular interest to us, is based on hyperbox fuzzy sets originally introduced by Simpson (1993) in the 1990s and then later improved, extended, and generalized by Gabrys and Bargiela (2000), Gabrys (2002a,b, 2004) as well as a large number of other researchers (Khuat et al., 2021). In its original paper, Simpson (1993) introduced the Fuzzy Min-Max (FMM) algorithm as an unsupervised clustering method for pattern clustering. FMM is a neuro-fuzzy algorithm that integrates fuzzy inference systems and adaptive neural networks. Employing a fuzzy inference system facilitates the creation of a neuro-fuzzy system capable of handling uncertainty. Additionally, using an adaptive neural network structure allows one to use learning approaches to find optimal parameters (Cortés-Antonio et al., 2020). Furthermore, having the ability to extract fuzzy if-then rules from the network architecture means that it is no longer a black box model. As mentioned, unsupervised clustering methods can partition an input space in neural network-based regression (e.g., RBF) models. Hence, some examples of FMM-based regression models have also been found.

Simpson and Jahns (1993) introduced an FMM-based framework for function approximation. In their approach, the authors utilized the FMM clustering method to partition the input feature space and used the hyperbox fuzzy sets representing clusters and the associated trapezoidal fuzzy membership functions as basis functions to estimate the target output. Similarly to the RBF networks, the output was a weighted combination of hyperbox fuzzy sets membership values (Simpson and Jahns, 1993). In another study, Tagliaferri et al. (2001) developed an innovative FMM-based model for function approximation, enhancing the FMM clustering algorithm for better feature space partitioning. The authors asserted that batch learning algorithms, which partition the feature space using the entire dataset, help eliminate the dependence on the order of data presentation. According to Tagliaferri et al. (2001), this adjustment significantly enhanced the model's performance.

Additionally, Brouwer (2005) proposed a novel automatic learning algorithm for the FMM-based function approximation model. Given that the loss function of the FMM is not differentiable and thus incompatible with gradient descent optimization, Brouwer (2005) implemented a helper neural network to approximate the loss function. This network allowed the application of the gradient descent algorithm to adjust the network parameters. Brouwer's new learning framework demonstrated the capability to achieve superior optimal values compared to conventional training algorithms. These RBF-like models involve clustering-based input feature space partitioning, which helps manage high-dimensional input data and overcome a curse of

dimensionality problems quite common in other neuro-fuzzy regression methods like ANFIS (Jang, 1993).

Despite some initial interest, success, and attractive features, far fewer hyperbox-based regression models exist than their classification/clustering counterparts (Khuat et al., 2021). In this paper, we introduce a novel neuro-fuzzy structure and learning procedure called Hyperbox Mixture Regression (HMR) that utilizes the hyperbox input space partitioning to overcome the curse of dimensionality problem while taking full advantage of its universal approximator capabilities and model transparency. The proposed method employs the hyperbox representations in constructing the basis functions in the first layer. It is worth noting that the new HMR structure is more straightforward than conventional neuro-fuzzy structures like ANFIS (Jang, 1993) and has a lower computational complexity to produce the output.

A hyperbox is a convex n-dimensional box in the feature space that assigns a full membership value to patterns within it (Simpson, 1992; Khuat et al., 2021) and is defined by its maximum and minimum points. The utilization of hyperboxes for the input space partitioning offers notable advantages. A key benefit is their ability to learn from an input sample in a single-pass process, leading to a significant boost in the learning speed of the system (Khuat and Gabrys, 2021b; Gabrys and Bargiela, 2000; Gabrys, 2002a). In addition, the single-instance-based hyperbox generation makes the model more robust to limited samples in imbalanced datasets. Moreover, the hyperbox learning framework is free from limitations such as being trapped in a local minimum or divergence due to the presence of outliers. Additionally, employing hyperboxes increases transparency (Dandl et al., 2023) and enables the system to identify essential basis functions, thereby reducing overall complexity, particularly in high-dimensional spaces.

The ability of the HMR model to infer and explain under uncertain conditions using the generated hyperbox fuzzy sets is crucial in industrial fields such as mAb production, where measurement noise is frequently associated with process parameters. This paper will apply the proposed HMR model to predict the performance of mAb production processes based on the critical process parameters used as input features (Gangadharan et al., 2021). The empirical dataset encompasses information from 106 bioreactors, capturing biological parameters over 15 cell culture days. The model's primary objective is to predict values of key performance indicators, such as Viable Cell Density (VCD) and mAb concentration, for antibody production processes within the subsequent two days. Consequently, the development involves the creation of two distinct predictive models: one for forecasting antibody production one day ahead and another for predicting the production two days ahead. The main contributions of this paper can be summarized as follows:

1. Introducing a new neuro-fuzzy model structure for bioprocess performance prediction to address the limited samples, high-dimensional data space, and transparency limitations under uncertain conditions.
2. Introducing a novel learning procedure that learns in a single input data pass process and generates essential basis functions for regression, effectively increasing the learning rate and reducing system complexity in high dimensional problems.
3. Employing a dynamically weighted combination of local linear regressors associated with each hyperbox to increase accuracy and decrease network complexity, especially in nonlinear problems.
4. Introducing a normalization layer in the proposed structure to reduce the risk of numerical instability in the next layers.
5. Forecasting the key performance indicators of antibody production processes over the next two cell culture days.

The rest of the paper is organized as follows. The next section describes in detail and illustrates the proposed model structure and training algorithm. In the third section, comprehensive experiments evaluate the proposed model in different scenarios. Finally, section four concludes the paper and highlights the key contributions.

2. Proposed method

In this section, the proposed model is described in detail. The section is divided into two parts representing the proposed model structure and learning procedure.

2.1. Hyperbox mixture regression structure

Fig. 1 illustrates the proposed HMR structure. According to Fig. 1, the HMR consists of four layers. In the first layer, each node represents a fuzzy hyperbox and computes membership values for the input sample. The number of nodes in the first layer is determined dynamically during the learning stage. The second layer normalizes the computed membership values and reduces the risk of numerical instability that can arise in the subsequent layers. In the third layer, we utilize a linear regressor for each hyperbox from the previous layer. The final network output is computed using the sum of the weighted, local (i.e. associated with each hyperbox) linear regressions. The following equations compute the system's output based on this structure.

Layer 1: u_{HB_l} for $l = 1, 2, \dots, L$

$$u_{HB_l}(X_h) = \min_{i=1,\dots,n} (\min([1 - g(x_{hi} - w_{li}, \lambda_i)], [1 - g(v_{li} - x_{hi}, \lambda_i)])) \quad (1)$$

where $i = 1, 2, \dots, n$ shows the feature index, L is the number of hyperboxes, u_{HB_l} represents the membership value of the lth node in the interval $[0, 1]$, v_{li} and w_{li} show minimum and maximum of the hyperbox points in the feature space, $X_h = (x_{h1}, x_{h2}, \dots, x_{hn})$ represents the hth input sample, n shows the number of input variables, and $g(r, \lambda_i)$ can be computed using the following Eq. (2):

$$g(r, \lambda_i) = \begin{cases} 1 & \text{if } r\lambda_i > 1 \\ r\lambda_i & \text{if } 0 \le r\lambda_i \le 1 \\ 0 & \text{if } r\lambda_i < 0 \end{cases} \quad (2)$$

Where λ_i is a sensitivity coefficient for the hyperbox. Fig. 2 illustrates the hyperbox membership function for two input features. According to the figure, the membership function assigns a value between 0 and 1, indicating each sample's membership value. Decisions are made based on these membership values. The second layer is the normalization layer, which normalizes computed membership values.

$$\text{Layer 2: } \bar{Z}_{hl} = \frac{u_{HB_l}(X_h)}{\sum_{j=1}^{L} u_{HB_j}(X_h)} \text{ for } l = 1, 2, \dots, L \quad (3)$$

where $\bar{Z}_{hl}$ indicates the normalized membership value for the hth sample.

In the third layer, the f_{hl} function for each hyperbox is computed using Eq. (4):

$$\text{Layer 3: } f_{hl} = \sum_{i=1}^{n} d_{li} x_{hi} + r_l \text{ for } l = 1, 2, \dots, L \quad (4)$$

Where the d_{li} and r_l are the local linear regression function parameters and will be optimized during the learning stage. Please note that f_{hl} can take different forms such as $f_{hl} = r_l$, in which case the output would be a sum of the weighted fuzzy membership values to locally trainable highly nonlinear functions. However, these other forms of f_{hl} are outside the scope of the current paper.

The output of the system can be obtained using Eq. (5):

$$\text{Layer 4: } y_h = \sum_{l=1}^{L} f_{hl} \bar{Z}_{hl} \quad (5)$$

Where y_h is the system's output, the HMR parameters must be optimized using a learning procedure to ensure accurate estimation. The following section introduces a new learning procedure for the proposed HMR structure.

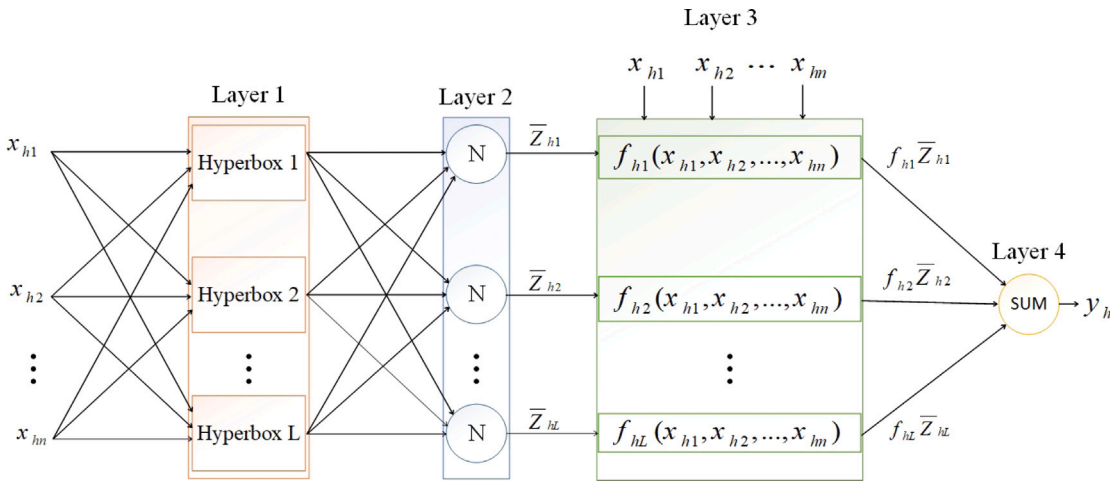

Fig. 1. Proposed HMR structure with L radial functions.

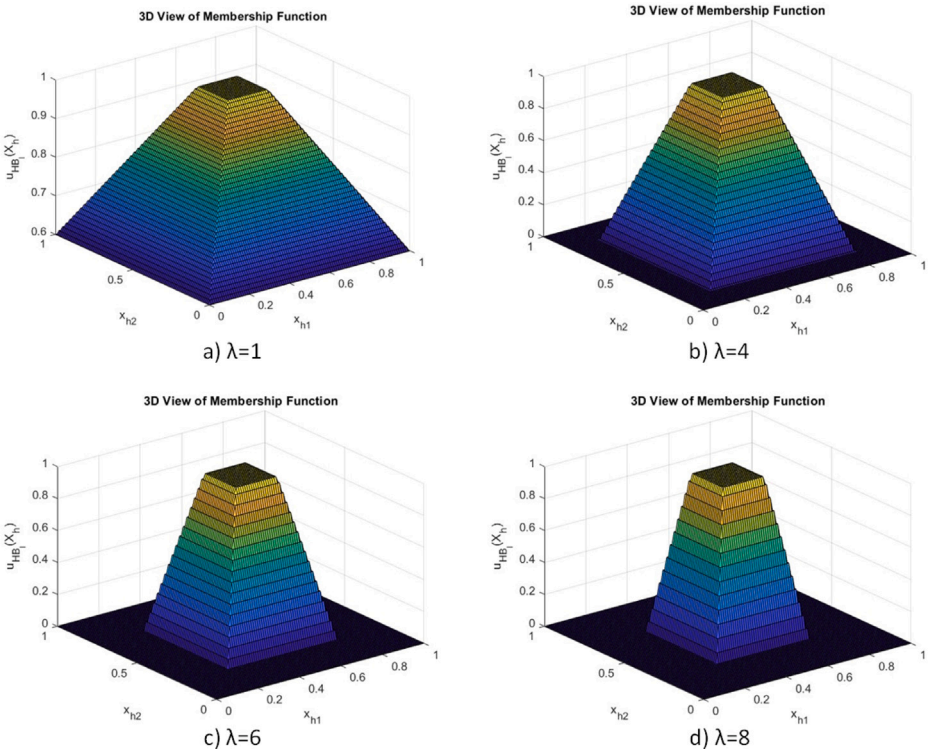

Fig. 2. 3D view of the membership function for different λ.

2.2. Hyperbox mixture regression learning procedure

This section describes a novel learning procedure to optimize the HMR parameters. The learning algorithm requires a single input data pass process for feature space partitioning, significantly increasing the learning speed. The proposed HMR learning algorithm comprises two stages: Hyperbox Min-Max clustering and the Least Squares Optimization (LSO) (see Fig. 3).

Hyperbox min–max clustering involves creating hyperboxes and producing the first layer of the model. In the second stage, the least squares optimization finds the local f_{hl} regressors parameters. The rest of this section describes the proposed learning procedure in detail.

2.2.1. Hyperbox min-max clustering

As mentioned before, the hyperboxes are created and form the first layer of the network. This stage involves:

1. Computing membership values
2. Selecting the top-K winning hyperboxes
3. Checking the expandability
4. Expanding the selected winning hyperbox

Consider an input sample, Eq. (1) is used to compute the sample's membership values and then finding the winning hyperbox with the highest membership value. Then, the algorithm expands the winning hyperbox to contain the input sample. If the winning hyperbox does not meet the expansion criterion, the algorithm checks the next winner until the Kth winner. If all K winning hyperboxes do not meet the expansion criterion, the algorithm generates a new hyperbox containing the input sample.

The following equation indicates the expansion criterion for each dimension i of the lth hyperbox:

$$\theta \geq \max(w_{li}, x_{hi}) - \min(v_{li}, x_{hi}), \tag{6}$$

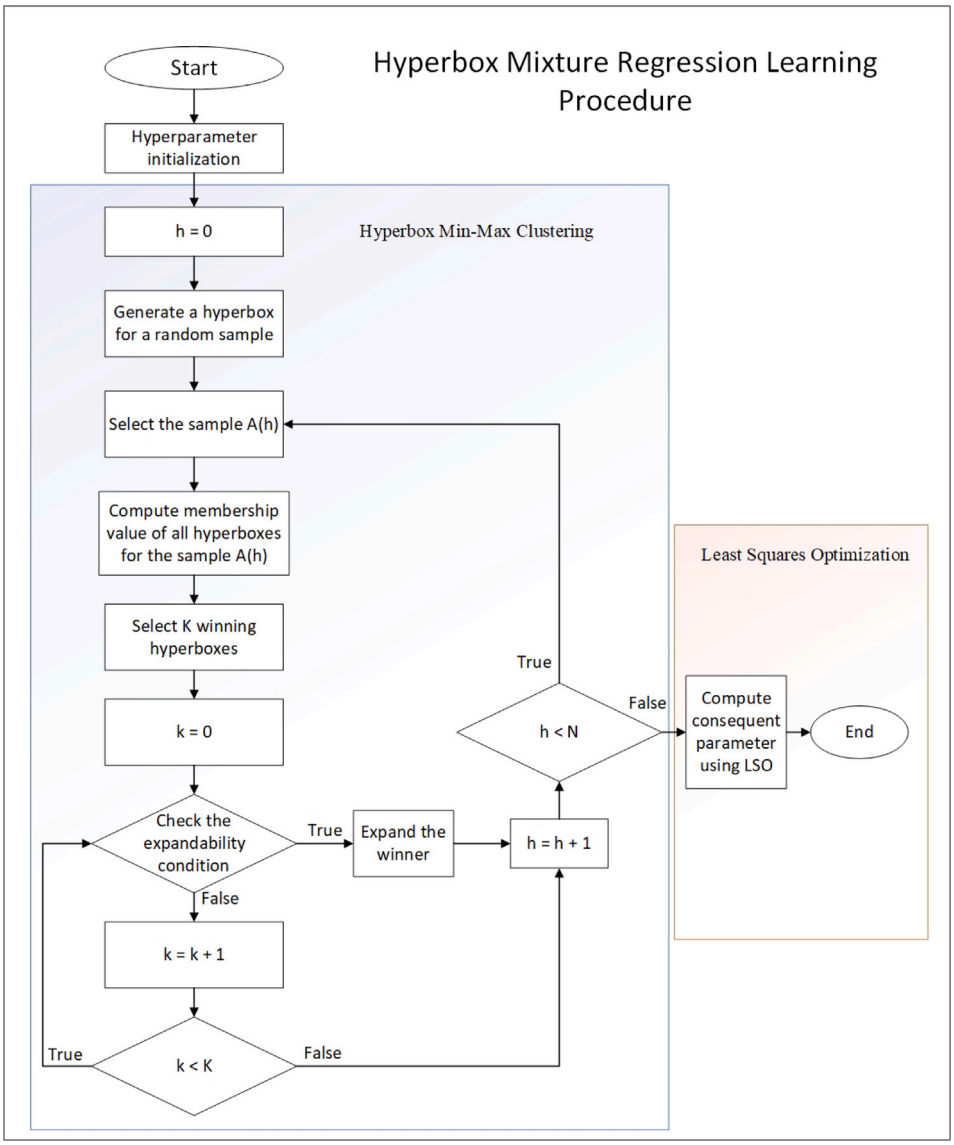

Fig. 3. Flowchart of the proposed HMR learning procedure.

where θ is the expansion coefficient ranging between $[0, 1]$. It will be expanded if the expansion criterion is satisfied for 60% of the dimensions within the winning hyperbox (Kumar et al., 2019). This criterion leads to generating fewer hyperboxes, and as a result, the system will have lower complexity (Kumar et al., 2019). The following equations illustrate the expansion formula:

$$v_{li}^{new} = \min(v_{li}^{old}, x_{hi}) \tag{7}$$

$$w_{li}^{new} = \max(w_{li}^{old}, x_{hi}) \tag{8}$$

This algorithm creates the necessary hyperboxes for the first layer. Therefore, this stage generates the required basis functions for regression. The number of created basis functions can be adjusted using the expansion coefficient θ. In other words, when we choose a large value of θ, the number of created basis functions decreases. On the other hand, a smaller value of θ can lead to the system creating more basis functions. It is essential to highlight that selecting a small value for θ is preferable when dealing with a highly nonlinear target. Conversely, opting for a larger value of θ is suitable for less complex functions.

2.2.2. Least squares optimization

This stage aims to optimize the local regressor parameters. These parameters can be optimized using the LSO algorithm. Consider the following equation:

$$y_h = \sum_{l=1}^{L} f_{hl}\bar{Z}_{hl} = \sum_{l=1}^{L}\left(\sum_{i=1}^{n}(\bar{Z}_{hl}d_{li}x_{hi}) + \bar{Z}_{hl}r_l\right) \tag{9}$$

Eq. (9) can be rewritten as the following equation:

$$Y = AD \tag{10}$$

where D is an $S\times1$ vector that contains the local regressors parameters. A represents an $N \times S$ matrix described as in Eq. (11). In addition, Y shows an $N \times 1$ vector that contains the output targets. Furthermore, S and N indicate the total number of local regressor parameters and input samples.

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1L} \\ a_{21} & a_{22} & \cdots & a_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NL} \end{bmatrix} \tag{11}$$

$$a_{hl} = \left[\bar{Z}_{hl}x_{h1}, \bar{Z}_{hl}x_{h2}, \ldots, \bar{Z}_{hl}x_{hn}, \bar{Z}_{hl}\right] \tag{12}$$

$$D = \left[d_{11}, \ldots, d_{1n}, r_1, \ldots, d_{L1}, \ldots, d_{Ln}, r_L\right]^T \tag{13}$$

$$Y = \left[y_1, y_2, \ldots, y_N\right]^T \tag{14}$$

Optimal values for D can be computed by optimizing the following square loss function:

$$\mathcal{L}_s = \|AD - Y\|^2 \tag{15}$$

Therefore, optimal values for D can be obtained using Eq. (16):

$$D = (A^T A)^{-1} A^T Y \tag{16}$$

where A^T shows the transpose of the matrix A, and $(A^T A)^{-1}$ indicates the pseudo-inverse of A.

3. Experimental results

The experiments in this paper were performed on a bioprocess dataset given in Gangadharan et al. (2021) to evaluate the efficacy of the learning models. The dataset encompasses a variety of operational scales, from bench-top to manufacturing levels, reflecting the diverse bioprocess environments encountered in the real-world scenarios (Gangadharan et al., 2021). Therefore, the dataset ensures that the proposed model accurately represents the complexities of real-world bioprocessing scenarios. The dataset comprises data from 106 cultures and encompasses 23 critical process parameters per culture. These parameters include mAb, Viable cell density (VCD), Elapsed Culture Time (ECT), Elapsed Generation Number (EGN), Total cell density (TCD), potassium concentration (K^+), etc. The dataset spans a collection period of 15 days for each culture. The objective of the dataset is to predict the values of mAb and VCD for the upcoming two days. However, the dataset presents a challenge with 3074 missing values filled using the missing value imputation method presented in Gangadharan et al. (2021).

The proposed model addresses the dimensionality curse associated with popular fuzzy-based models such as ANFIS (Jang, 1993). Therefore, the experiments are designed to evaluate the model in both high-dimensional and low-dimensional scenarios and compare the proposed HMR model with the ANFIS Hybrid Learning (HL) and Fuzzy Neural Network (FNN) Back-Propagation (BP) algorithms (Nik-Khorasani et al., 2024). The comparison focuses on model complexity, learning time, and model performance. Predictive models were developed to forecast future mAb concentration and VCD values based on the current day's process parameter inputs.

This paper implements the ANFIS model described in Nik-Khorasani et al. (2024) and the FNN model outlined in Rafiei and Akbarzadeh-T (2022). The architecture of the ANFIS model comprises five layers: fuzzification, rule antecedent, normalization, rule consequent, and rule inference. In contrast, the FNN model consists of three layers: fuzzification, rule layer, and rule inference layer. Both models utilize Gaussian membership functions for the input variables. All methods were implemented in Python, and we applied the same preprocessing steps to both the ANFIS and FNN models as those implemented in our HMR model.

The generalization of predictive algorithms is assessed using the 5-fold cross-validation method. Therefore, 106 cultures were divided into five folds; four were used for training, and the remaining fold was used for testing. This training and testing process was repeated for all five folds, and the reported results are the mean of five experiments corresponding to each testing fold. This approach evaluates how well the predictive models generalize in predicting the mAb concentration and VCD values for the cultures the trained model has not previously encountered. Consider the following vector:

$$C_i(t) = [ECT_i(t), VCD_i(t), \ldots, Glutamine_i(t)]_{1\times 23}, \tag{17}$$

where $C_i(t)$ indicates the ith culture that contains 23 process parameters (features) in the respective tth day (t is a discrete variable). Consider $C_1(t)$ as an example, for VCD prediction, $C_1(t)$ corresponds to the first culture with 23 input features on the respective tth day, where $1 \leq t \leq 15$. The $C_1(t)$ target is $VCD_1(t+1)$, representing the VCD value at the day $t+1$ for the first culture. Additionally, for mAb values, the target follows a similar pattern; for instance, the target for $C_i(t)$ would be $mAb_i(t+1)$.

In the rest of this section, comprehensive experiments are conducted to evaluate and compare the proposed model with other competing models. The first experiment (Section 3.1) evaluates models' performance for one-day-ahead mAb and VCD prediction in high-dimension scenarios, including all 23 input features. The second experiment (Section 3.2) uses a feature selection algorithm to reduce the input feature space. The third experiment (Section 3.3) compares alternative models' accuracy, complexity, and learning rate for one-day-ahead mAb and VCD prediction in low dimensional scenarios after feature selection. In the last part (Section 3.4), comprehensive experiments are conducted to evaluate the HMR model for one-day-ahead and two-day-ahead mAb and VCD prediction. The λ_i is set to 1 in all following experiments.

3.1. High dimensional scenario

We trained the predictive models using all 23 input features in the first part. The 5-fold cross-validation method was used to assess the performance of the trained models. In addition, the hyperparameter θ is set to 0.1. Therefore, for each of the five experiments, we have:

- Training dataset: all $C_i(t)$ where i are cultures within the four training folds, and $1 \leq t \leq 14$
- Testing dataset: all $C_i(t)$ where i are cultures within the testing fold, and $1 \leq t \leq 14$
- The targets of datasets are $VCD_i(t+1)$ and $mAb_i(t+1)$.

As mentioned earlier, utilizing the hyperbox concept increases the transparency of the model. In the following example, the generated rules for each hyperbox are illustrated for each input pattern $x_h = (x_{h1}, \ldots, x_{hn})$:

Rule 1: If $v_{11} \leq x_{h1} \leq w_{11}$ AND … AND $v_{1n} \leq x_{hn} \leq w_{1n}$,
then $f_{h1} = d_{11}x_{h1} + d_{12}x_{h2} + \cdots + d_{1n}x_{hn} + r_1$

Rule 2: If $v_{21} \leq x_{h1} \leq w_{21}$ AND … AND $v_{2n} \leq x_{hn} \leq w_{2n}$,
then $f_{h2} = d_{21}x_{h1} + d_{22}x_{h2} + \cdots + d_{2n}x_{hn} + r_2$

⋮

Rule L: If $v_{L1} \leq x_{h1} \leq w_{L1}$ AND … AND $v_{Ln} \leq x_{hn} \leq w_{Ln}$,
then $f_{hL} = d_{L1}x_{h1} + d_{L2}x_{h2} + \cdots + d_{Ln}x_{hn} + r_L$

The parameters such as v_{Li}, w_{Li}, d, and r should be optimized in the training stage.

Table 1 illustrates the results of predictive models for all input variables using 5-fold cross-validation. Due to the curse of dimensionality, the original ANFIS models pose a significant computational burden when utilizing 23 input variables. For instance, if the ANFIS creates only two membership functions for each input feature, the grid partitioning leads to the generation of 2^{23} rules, leading to challenges in optimization. This limitation necessitates selecting only one membership for each input variable, consequently diminishing the performance of the original ANFIS models. In contrast, HMR dynamically generates basis functions without such constraints. Therefore, HMR significantly outperforms the ANFIS HL algorithm and FNN BP in the high-dimensional scenario.

Table 1
The testing RMSE and standard deviation scores of different predictive models using all 23 input features over 5-fold cross-validation.

Model	$VCD_i(t+1)$		$mAb_i(t+1)$	
	Training RMSE	Testing RMSE	Training RMSE	Testing RMSE
ANFIS HL	0.3319 ± 0.0074	0.3321 ± 0.0494	0.2553 ± 0.0034	0.2815 ± 0.0382
FNN BP	0.1711 ± 0.0048	0.1727 ± 0.0285	0.1598 ± 0.0009	0.1606 ± 0.0121
HMR	0.0283 ± 0.0010	0.0537 ± 0.0238	0.0296 ± 0.0017	0.0516 ± 0.0110

Table 2
Correlation scores of the first six important input features with the $mAb(t+1)$ target.

Training	$mAb(t)$	ECT	$Glutamine$	K^+	EGN	$Osmolality$
Fold-1	0.97	0.78	0.77	0.66	0.65	0.53
Fold-2	0.97	0.77	0.76	0.67	0.62	0.54
Fold-3	0.97	0.80	0.76	0.65	0.64	0.55
Fold-4	0.97	0.80	0.76	0.67	0.68	0.55
Fold-5	0.97	0.78	0.76	0.64	0.65	0.53

Table 3
Correlation scores of the first six important input features with the $VCD(t+1)$ target.

Training	$VCD(t)$	TCD	$Temperature$	pCO2	$HCO3^-$	Na^+
Fold-1	0.95	0.83	−0.59	−0.43	−0.41	0.35
Fold-2	0.95	0.83	−0.59	−0.37	−0.35	0.38
Fold-3	0.95	0.83	−0.63	−0.39	−0.36	0.33
Fold-4	0.95	0.83	−0.61	−0.35	−0.33	0.34
Fold-5	0.95	0.83	−0.61	−0.36	−0.35	0.35

3.2. Feature selection

We conducted a feature selection procedure to reduce the input feature space in the following experiments. The feature selection algorithm is primarily based on correlation metrics to effectively address multicollinearity among quantitative parameters. The correlation matrix enables the identification and mitigation of feature redundancy, which is crucial for ensuring the robustness of the trained models.

Reducing the feature space has several advantages. It can improve model performance by enabling the model to focus on the most relevant information. Selecting only the most essential features can enhance prediction accuracy. In addition, it reduces the complexity of the trained model. A lower dimensionality simplifies the model, making it easier to interpret and manage. This reduction can also help to avoid overfitting, where the model learns noise rather than the underlying pattern. Furthermore, it decreases the training time. While not the sole reason, reducing the number of features can lead to faster training, as the model has fewer variables to process. Additionally, lowering the number of features helps to mitigate the curse of dimensionality problems. While reducing model training time is a benefit, the primary goals are improving model performance and managing complexity effectively.

The feature selection algorithm utilized 5-fold cross-validation to identify the most critical input features within the training data. The original data was split into five folds, with four folds used for feature selection, and one was reserved for validation of the selected features (outer loop). We employed a separate inner 5-fold cross-validation from each above outer training fold for hyper-parameter tuning, using one fold for model validation and the remaining samples were used for model training to select the best hyper-parameter configuration. This ensured that the selected features were independent of the validation data, thereby providing reliable results with the necessary generalization.

Next, we reordered the input variables based on their correlation scores with the target in the training data. It was assumed that input variables with higher correlations to the target would contribute more significantly to the output. In the arranged dataset, the first variable exhibited the highest correlation, while the last showed the lowest. Tables 2 and 3 illustrate the six variables with the highest correlations with the targets for each training fold, where correlations close to 1 or −1 indicate strong relationships with the selected features.

In the feature selection algorithm, the predictive model was trained using the first feature with the highest correlation value, and the RMSE on the outer validation fold was computed. The HMR model was chosen as the predictor in the feature selection process due to its rapid learning speed and dynamic basis function generation. However, because the used model includes one hyperparameter θ, which required optimization, at the same time, we simultaneously performed a parameter tuning procedure with values ranging from 0.1 to 0.7 to identify the optimal θ for each training fold. During this stage, we selected the θ value that yielded the lowest average RMSE over all five inner validation folds. Then, the best selected hyper-parameter setting was used to train the model on the outer training fold with the current set of selected features. Subsequently, the following input features were added sequentially, with their respective RMSE on the outer validation fold calculated. This process was repeated for all 23 input variables, and the top-k features were identified as the most critical for training the predictive model, where k represents the number of features that resulted in the lowest RMSE on the outer validation fold. Table 4 shows the RMSE for each outer validation fold.

Table 4 illustrates the number of input variables selected in each training fold to reach the best performance on the corresponding validation fold of the feature selection algorithm. Based on the obtained results, we selected the set of features that appeared in the selected list in at least three out of the five outer validation folds as the final set of best-selected features. Consequently, the first eight variables for VCD and the first seven for mAb with the highest correlation with the targets were the best selections that can lead to low validation RMSE scores for most validation folds. Results suggested that adding extra variables did not substantially enhance accuracy; instead, it increased model complexity and learning time and could potentially diminish the predictive model's performance. Finally, input variables of the mAb prediction for each culture are illustrated in Eq. (18):

$$C_i'(t) = [mAb_i(t), ECT_i(t), Glutamine_i(t), K_i^+(t), EGN_i(t), Osmolality_i(t), ACV_i(t)]_{1\times 7} \tag{18}$$

with the target of $mAb_i(t+1)$. Similarly, input variables for the VCD prediction are:

$$C_i''(t) = [VCD_i(t), TCD_i(t), Temperature_i(t), Na_i^+(t), pCO2_i(t), HCO3_i^-(t), EGN_i(t), ECT_i(t)]_{1\times 8} \tag{19}$$

with the target of $VCD_i(t+1)$. It can be observed that the mAb concentration and VCD of the current day have a significant impact on the predictive performance of the following day. Regarding other process parameters, our feature selection technique accurately identifies the relevant parameters regarding their biochemical significance for cell growth and mAb concentration in cell culture bioreactors. For instance, glutamine is an essential supplement for sustaining cell growth and product concentration (Sheikholeslami et al., 2014; Pérez-Rodriguez et al., 2020). Salts such as sodium (Na^+) and potassium (K^+) play a crucial role in various cellular processes, including transmembrane potential, nutrition, buffering, osmolality, and signal transduction (Ritacco et al., 2018). These processes, in turn, affect cell growth and mAb titer. Meanwhile, high values of temperature, partial pressure of carbon dioxide (pCO2), and bicarbonate ($HCO3^-$) can negatively influence

Table 4
The best validation RMSE and top-K input features for the VCD and mAb prediction using 5-fold cross validation.

Val fold	$\widehat{VCD}_i(t+1)$		$\widehat{mAb}_i(t+1)$	
	Best Val RMSE	Best top-K features	Best Val RMSE	Best top-K features
1	0.0372 ± 0.0048	8	0.0431 ± 0.0077	6
2	0.0410 ± 0.0052	8	0.0432 ± 0.0035	11
3	0.0392 ± 0.0103	8	0.0427 ± 0.0057	5
4	0.0415 ± 0.0058	9	0.0435 ± 0.0056	6
5	0.0409 ± 0.0055	9	0.0395 ± 0.0095	8

the cell growth, metabolism, and mAb productivity (Goudar et al., 2007; Martínez et al., 2015), and they need to be controlled within acceptable ranges. Total cell density (TCD) also positively correlates with VCD in the cell culture process (Gangadharan et al., 2021). In the literature, culture osmolality has consistently shown a positive effect on mAb productivity and titer (Alhuthali et al., 2021). In the cell culture process, cell culture time (ECT) and the number of cell divisions (represented by the EGN feature) can significantly impact mAb concentration and overall cell growth. Antibody concentration might be low in the early phases of cell culture because of low cell densities. As the culture progresses, cells multiply, and the mAb concentration typically increases. However, when culture time is prolonged, cell growth might slow down due to overcrowding, nutrient depletion, or accumulation of inhibitory metabolites, decreasing mAb production and cell viability. Similarly, higher numbers of cell divisions initially result in increased cell numbers and mAb concentration. However, with continued divisions, cells might experience replicative senescence, decreasing viability, and mAb yield.

Using the selected feature subset in Eq. (18), the training and testing sets for each experiment are defined as follows:

- Training dataset: all $C_i'(t)$ where i are cultures within the four training folds, and $1 \leq t \leq 14$
- Testing dataset: all $C_i'(t)$ where i are cultures within the testing fold, and $1 \leq t \leq 14$
- The target of dataset is $mAb_i(t+1)$.

The VCD training and testing datasets are produced similarly to mAb. The predictive models aim to predict the future values for the mAb concentration and VCD, and they are defined as follows:

$$\widehat{mAb}_i(t+1) = \mathcal{F}(C_i'(t)) \tag{20}$$

$$\widehat{VCD}_i(t+1) = \mathcal{G}(C_i''(t)), \tag{21}$$

where $\mathcal{F}$ and $\mathcal{G}$ are the trained models and $\widehat{mAb}_i$ and $\widehat{VCD}_i$ shows the predicted values. The best prediction would be the prediction that minimizes the following equations:

$$\mathcal{L} = \|\widehat{mAb} - mAb\|^2 \tag{22}$$

$$\mathcal{L} = \|\widehat{VCD} - VCD\|^2, \tag{23}$$

Where mAb and VCD are actual values of cell cultures. The learning algorithms optimize parameters to minimize $\mathcal{L}$ in the above equations. The experiments in the following subsection utilize the selected features to train the predictive models for one-day-ahead mAb and VCD predictions and compare the various learning models in terms of accuracy, complexity, and learning time.

3.3. Low dimensional scenario

In this section, experiments compare accuracy, complexity, and learning time between the competing predictive models in the low-dimensional scenario. Predictive models aim to predict one-day-ahead values of the mAb concentration and VCD. In this part, we employed the dataset together with features selected in the previous subsection.

Table 5
mAb prediction RMSE and number of neurons in the first layer for predictive models using the selected input features and 5-fold cross validation.

Predictive model	ANFIS HL	FNN BP	Proposed model
1st layer size	128 ± 0	128 ± 0	32 ± 6
Training time	145.2 s	105.8 s	5.8 s
Tuning time	107.2 min	84.9 min	6.5 min
Train RMSE	0.0303 ± 0.0021	0.0561 ± 0.0091	0.0392 ± 0.0016
Test RMSE	0.0989 ± 0.0390	0.0701 ± 0.0298	0.0455 ± 0.0062

Table 6
VCD prediction RMSE and number of neurons in the first layer for predictive models using the selected input features and 5-fold cross validation.

Predictive model	ANFIS HL	FNN BP	Proposed model
1st layer size	256 ± 0	256 ± 0	133 ± 2
Training time	605.3 s	91.6 s	44.4 s
Tuning time	230.7 min	107.1 min	6.0 min
Train RMSE	0.0279 ± 0.0019	0.0722 ± 0.0151	0.0348 ± 0.0017
Test RMSE	0.0680 ± 0.0126	0.0755 ± 0.0144	0.0409 ± 0.0060

To enhance the evaluation process, we initially optimized hyperparameters for each model. The optimization of hyperparameters was performed using a grid search method (Pedregosa et al., 2011) with 5-fold cross-validation on each training fold. In adherence to this approach, predictive models undergo training using various combinations of hyperparameters, and the set of hyperparameters yielding the lowest average RMSE on all validation folds is selected as the optimal configuration. After hyperparameter optimization and model training, the values of training and testing RMSE, the number of generated neurons in the first layer, training time, and tuning time for hyperparameter optimization are compared among different models.

Tables 5 and 6 illustrate the best performance of predictive models for one-day-ahead mAb and VCD prediction. The results suggest that, in these experiments, HMR produces significantly fewer neurons (i.e. hyperboxes equivalent to the fuzzy if-then rules in other methods) than the competing models to reach its best performance, signifying its lower complexity than other models. Additionally, the HMR model achieves lower test RMSE value, suggesting superior accuracy compared to other predictive models. Furthermore, the proposed model exhibits significantly higher learning speeds than other alternative models in training and tuning processes. Moreover, the training and testing RMSE scores in the proposed model are close, implying its good generalization. Notably, the testing accuracy of all predictive models using the selected feature subset is significantly higher than when using all 23 input features, as depicted in Table 1.

According to the above outcomes, the proposed HMR model significantly outperforms the conventional ANFIS models. The proposed model offers lower complexity, a higher learning rate, and accuracy with no curse of dimensionality limitation. Results also confirm that the proposed model performs better than conventional ANFIS in low and high-dimensional problems. Next, we assess the proposed model for one-day-ahead and two-day-ahead mAb and VCD prediction.

3.4. Analyses of HMR in the process performance prediction

This section involves comprehensive experiments to assess the proposed HMR model's ability to predict mAb and VCD one and two days

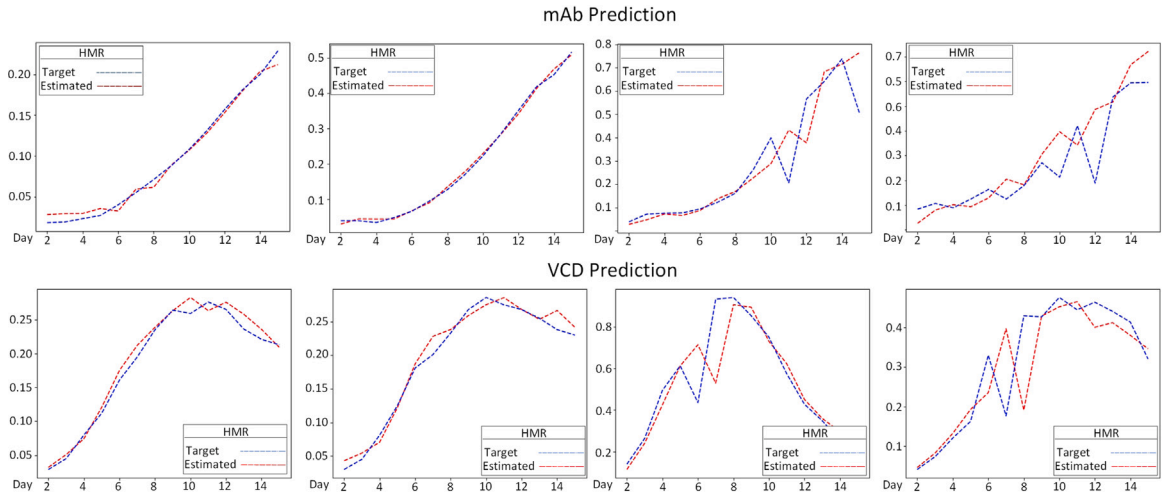

Fig. 4. One-day-ahead prediction of the mAb concentration and VCD using HMR for representative bioreactors. The best predictions are illustrated in the two left figures, while the worst predictions are depicted in the two right figures.

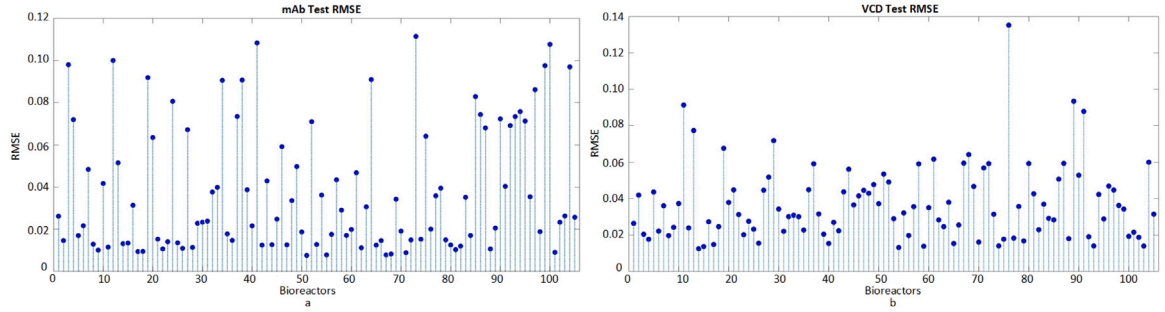

Fig. 5. Testing RMSE of (a) mAb and (b) VCD one-day-ahead prediction for 106 bioreactors using HMR.

ahead. In this section, the proposed model employs hyperparameters and features selected in previous parts. We trained the model using 5-fold cross-validation for a better assessment and computed the RMSE of all 106 bioreactors during their respective 14 days.

Fig. 4 shows the best and the worst mAb and VCD predictions using the HMR model. In addition, Fig. 5 shows the testing RMSE scores of 106 bioreactors across their respective 14 culture days. Results demonstrate that the proposed model can track the target accurately. However, the performance of predictors is reduced for cultures experiencing sudden increases or decreases in the mAb concentration and VCD on the next culture day compared to the current day, as depicted in the worst-case scenarios in Fig. 4. This inconsistency might arise from operator interventions in the cell culture process, such as adding specific nutrients (e.g., glucose) not included in the features. Additionally, Table 7 illustrates that the proposed model can more accurately predict the first five days of the cell culture process, while the predictive performance remains consistent for the subsequent culture days. However, the HMR model exhibits more significant variation in predicting mAb concentrations among 106 cultures from the 6th culture day compared to the initial five days. This variation is evident through higher standard deviation values in Table 7.

In addition, another HMR model was trained to predict two-day-ahead mAb and VCD values. We employed C_i^3 and C_i^4 to train the predictive model for two-day-ahead prediction using 5-fold cross-validation in the following experiments. Therefore, the mAb and VCD input variables are defined as follows:

$$C_i^3(t) = [mAb_i(t), ECT_i(t), Glutamine_i(t), K_i^+(t), EGN_i(t), Osmolality_i(t), ACV_i(t), mAb_i(t+1)]_{1\times 8} \quad (24)$$

with the target of $mAb_i(t+2)$. Similarly, input variables for VCD prediction are:

$$C_i^4(t) = [VCD_i(t), TCD_i(t), Temperature_i(t), Na_i^+(t), pCO2_i(t), HCO3_i^-(t), EGN_i(t), ECT_i(t), VCD_i(t+1)]_{1\times 9} \quad (25)$$

with the target of $VCD_i(t+2)$. The dataset for each experiment is presented as follows using these definitions:

- Training dataset: all $C_i^3(t)$ where i are cultures within the four training folds, and $1 \le t \le 13$
- Testing dataset: all $C_i^3(t)$ where i are cultures within the testing fold, and $1 \le t \le 13$
- The target of dataset is $mAb_i(t+2)$.

The VCD training and testing datasets are produced similarly to mAb. The following models are trained using the above equations:

$$\widehat{mAb}_i(t+2) = \mathcal{F}(C_i^3(t)) \quad (26)$$

$$\widehat{VCD}_i(t+2) = \mathcal{G}(C_i^4(t)), \quad (27)$$

The optimization algorithms aim to minimize the overall predicted values given in Eqs. (22) and (23) to determine optimal parameters of the predictive models. Additionally, it is essential to note that the second model requires the actual values of $mAb_i(t+1)$ and $VCD_i(t+1)$ as input features in the testing phase, which is impossible to obtain in reality. Consequently, during the testing phase of the trained model, we utilized the $\widehat{mAb}_i(t+1)$ and $\widehat{VCD}_i(t+1)$ instead of their actual values. Hence, the revised prediction equations for the model are presented as

Table 7
Mean testing RMSE of each day across 106 bioreactors for VCD and mAb.

Predicted day	mAb Test RMSE	VCD Test RMSE
Day 2	0.0125 ± 0.0179	0.0098 ± 0.0099
Day 3	0.0102 ± 0.0101	0.0152 ± 0.0172
Day 4	0.0130 ± 0.0159	0.0255 ± 0.0299
Day 5	0.0143 ± 0.0116	0.0247 ± 0.0252
Day 6	0.0210 ± 0.0220	0.0405 ± 0.0393
Day 7	0.0241 ± 0.0245	0.0419 ± 0.0569
Day 8	0.0248 ± 0.0300	0.0306 ± 0.0343
Day 9	0.0298 ± 0.0332	0.0287 ± 0.0274
Day 10	0.0356 ± 0.0437	0.0278 ± 0.0229
Day 11	0.0387 ± 0.0499	0.0308 ± 0.0274
Day 12	0.0381 ± 0.0584	0.0333 ± 0.0323
Day 13	0.0434 ± 0.0558	0.0248 ± 0.0235
Day 14	0.0365 ± 0.0504	0.0210 ± 0.0216
Day 15	0.0343 ± 0.0391	0.0231 ± 0.0212

Table 8
VCD and mAb two-day-ahead prediction testing RMSE and number of neurons in the first layer for the Proposed Model.

Predictive model	$\widehat{mAb}_i(t+2)$	$\widehat{VCD}_i(t+2)$
1st layer size	25 ± 1	129 ± 6
Training time	3.7 s	40.7 s
Tuning time	2.5 min	4.2 min
Train RMSE	0.0342 ± 0.0012	0.0348 ± 0.0014
Test RMSE	0.0390 ± 0.0051	0.0422 ± 0.0075

follows:

$$\widehat{mAb}_i(t+2) = \mathcal{F}(C_i'(t), \widehat{mAb}_i(t+1)) \tag{28}$$

$$\widehat{VCD}_i(t+2) = \mathcal{G}(C_i''(t), \widehat{VCD}_i(t+1)), \tag{29}$$

This model is utilizable in reality. Therefore, the first model predicts the next day's VCD and mAb values and another predictive model uses the prediction to estimate the VCD and mAb values for the next two days. Table 8 shows the proposed model performance for two-day-ahead prediction of the mAb concentration and VCD values. According to the results, the proposed model effectively generates critical basis functions, reducing complexity and accelerating the learning rate.

Our experiments demonstrate that the HMR model effectively addresses challenges related to high dimensionality and time-series data, significantly enhancing predictive accuracy and learning speed compared to traditional statistical methods. Nevertheless, there is potential for further improvement through various enhancement techniques. For example, incorporating methods to accelerate the training process could further boost the model's performance. By integrating techniques that identify and eliminate unsuitable hyperboxes during the expansion phase, the learning process of the HMR model can be streamlined (Khuat and Gabrys, 2021a). This optimization would reduce computational load and enhance efficiency, making the model more applicable for real-time bioprocess performance prediction.

Moreover, while physics-based models offer valuable insights, they often suffer from complexities that data-driven models can manage more effectively. Designing a hybrid model that combines both data-driven and physics-based approaches can leverage the strengths of each, ultimately enhancing the model's learnability (Zhang et al., 2019). For instance, by integrating a kinetic model with the HMR framework, one could leverage the strengths of both methodologies. The HMR model could be a data-driven layer that captures time-varying dependencies among parameters, while the kinetic model provides foundational process knowledge. This hybridization would enhance the model's accuracy and robustness, particularly in variable data quality and availability scenarios (Shah et al., 2022). This hybridization could improve predictive capabilities, allowing for more accurate monitoring and optimization of bioprocesses. By integrating these enhancements, the HMR model can achieve greater efficiency and accuracy while adapting more effectively to the complexities of real-world bioprocessing applications.

4. Conclusion

This paper introduced the HMR model as a novel approach to predicting bioprocess performance in monoclonal antibody production. Our findings demonstrate that the HMR model effectively addresses the challenges posed by high-dimensional time-series data, significantly improving predictive accuracy and learning speed compared to traditional statistical methods and other machine learning models.

The HMR model's unique ability to partition the input space using hyperboxes allows for enhanced interpretability and robustness, particularly in the face of uncertainty common in bioprocess data. Including hyperboxes offers advantages, such as learning data in a single-pass process and dynamic basis function generation. In addition, by employing a local linear regressor within each hyperbox, we achieved greater accuracy while reducing computational complexity, making the model suitable for real-time applications in biopharmaceutical production.

The effectiveness of the proposed method was assessed on the real-world mAb production problem with different scenarios. In our experimental study, we train the predictive models in both high and low dimensions for better evaluation. According to the empirical outcomes, the HMR can accurately estimate the output target in both scenarios. In addition, due to the generation of dynamic basis functions, the HMR only generates the critical basis functions, which leads to a less complex structure and avoids the curse of dimensionality problems.

However, the HMR is sensitive to the expansion coefficient θ. In other words, as the value of θ decreases, the HMR generates more hyperboxes, which increases complexity. On the other hand, increasing the expansion coefficient θ might affect the accuracy of the predictive models. In this paper, we employed grid-search for hyperparameter optimization. However, in future works, we aim to develop a learning procedure that adaptively select the θ value during the training stage. Additionally, to improve the model's performance in uncertain conditions, the multi-stream learning approach proposed in Yu et al. (2022) can be considered in the future.

Methods previously developed for robust hyperbox clustering systems exploiting various ensembling and method-independent statistical learning approaches (Gabrys, 2002b, 2004) will be explored and adopted to the regression settings considered in this paper. When applying the proposed method to predict the process performance in mAb production, only current values of process parameters were employed as input features to forecast the following day's culture. However, another model-building approach that incorporates actual values up to the current day within each process parameter as input features for early predictions regarding subsequent culture days is feasible and should also be considered in future studies. Additionally, incorporating more relevant features and employing non-linear feature selection methods could increase the model's accuracy and robustness to sudden process changes, which requires further investigation in future works.

CRediT authorship contribution statement

Ali Nik-Khorasani: Writing – original draft, Validation, Software, Methodology, Investigation, Data curation, Conceptualization. **Thanh Tung Khuat:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Conceptualization. **Bogdan Gabrys:** Writing – review & editing, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported under the Australian Research Council's Industrial Transformation Research Program (ITRP) funding scheme (project number IH210100051). The ARC Digital Bioprocess Development Hub is a collaboration between The University of Melbourne, University of Technology Sydney, RMIT University, CSL Innovation Pty Ltd, Cytiva (Global Life Science Solutions Australia Pty Ltd), and Patheon Biologics Australia Pty Ltd.

References

Alam, M.N., Anupa, A., Kodamana, H., Rathore, A.S., 2024. A deep learning-aided multi-objective optimization of a downstream process for production of monoclonal antibody products. Biochem. Eng. J. 208, 109357.

Alhuthali, S., Kotidis, P., Kontoravdi, C., 2021. Osmolality effects on CHO cell growth, cell volume, antibody productivity and glycosylation. Int. J. Mol. Sci. 22 (7), 3290.

Brouwer, R.K., 2005. Automatic training of a min-max neural network for function approximation by using a second feed forward network. Soft Comput. 9, 393–397.

Cortés-Antonio, P., Batyrshin, I., Martínez-Cruz, A., Villa-Vargas, L.A., Ramírez-Salinas, M.A., Rudas, I., Castillo, O., Molina-Lozano, H., 2020. Learning rules for sugeno ANFIS with parametric conjunction operations. Appl. Soft Comput. 89, 106095.

Dandl, S., Casalicchio, G., Bischl, B., Bothmann, L., 2023. Interpretable regional descriptors: Hyperbox-based local explanations. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, pp. 479–495.

de Campos Souza, P.V., 2020. Fuzzy neural networks and neuro-fuzzy networks: A review the main techniques and applications used in the literature. Appl. Soft Comput. 92, 106275.

Gabrys, B., 2002a. Agglomerative learning algorithms for general fuzzy min-max neural network. J. VLSI Signal Process. Syst. Signal, Image Video Technol. 32 (1), 67–82.

Gabrys, B., 2002b. Combining neuro-fuzzy classifiers for improved generalisation and reliability. In: Proceedings of the 2002 International Joint Conference on Neural Networks. Vol. 3, IEEE, pp. 2410–2415.

Gabrys, B., 2004. Learning hybrid neuro-fuzzy classifier models from data: to combine or not to combine? Fuzzy Sets and Systems 147 (1), 39–56.

Gabrys, B., Bargiela, A., 2000. General fuzzy min-max neural network for clustering and classification. IEEE Trans. Neural Netw. 11 (3), 769–783.

Gangadharan, N., Sewell, D., Turner, R., Field, R., Cheeks, M., Oliver, S.G., Slater, N.K., Dikicioglu, D., 2021. Data intelligence for process performance prediction in biologics manufacturing. Comput. Chem. Eng. 146, 107226.

Gangadharan, N., Turner, R., Field, R., Oliver, S.G., Slater, N., Dikicioglu, D., 2019. Metaheuristic approaches in biopharmaceutical process development data analysis. Bioprocess Biosyst. Eng. 42, 1399–1408.

Goudar, C.T., Matanguihan, R., Long, E., Cruz, C., Zhang, C., Piret, J.M., Konstantinov, K.B., 2007. Decreased pCO2 accumulation by eliminating bicarbonate addition to high cell-density cultures. Biotechnol. Bioeng. 96 (6), 1107–1117.

Jang, J.-S., 1993. ANFIS: adaptive-network-based fuzzy inference system. IEEE Trans. Syst. Man Cybern. 23 (3), 665–685.

Khuat, T.T., Bassett, R., Otte, E., Grevis-James, A., Gabrys, B., 2024. Applications of machine learning in antibody discovery, process development, manufacturing and formulation: Current trends, challenges, and opportunities. Comput. Chem. Eng. 108585.

Khuat, T.T., Gabrys, B., 2021a. Accelerated learning algorithms of general fuzzy min-max neural network using a novel hyperbox selection rule. Inform. Sci. 547, 887–909.

Khuat, T.T., Gabrys, B., 2021b. Random hyperboxes. IEEE Trans. Neural Netw. Learn. Syst..

Khuat, T.T., Ruta, D., Gabrys, B., 2021. Hyperbox-based machine learning algorithms: a comprehensive survey. Soft Comput. 25 (2), 1325–1363.

Kumar, S.A., Kumar, A., Bajaj, V., Singh, G.K., 2019. An improved fuzzy min–max neural network for data classification. IEEE Trans. Fuzzy Syst. 28 (9), 1910–1924.

Lai, P.-K., Gallegos, A., Mody, N., Sathish, H.A., Trout, B.L., 2022. Machine learning prediction of antibody aggregation and viscosity for high concentration formulation development of protein therapeutics. In: MAbs. Vol. 14, (1), Taylor & Francis, 2026208.

Lemke, C., Gabrys, B., 2010. Meta-learning for time series forecasting and forecast combination. Neurocomputing 73 (10–12), 2006–2016. http://dx.doi.org/10.1016/j.neucom.2009.09.020.

Li, T.-H.S., Kuo, P.-H., Chen, L.-H., Hung, C.-C., Luan, P.-C., Hsu, H.-P., Chang, C.-H., Hsieh, Y.-T., Lin, W.-H., 2020. Fuzzy double deep Q-network-based gait pattern controller for humanoid robots. IEEE Trans. Fuzzy Syst. 30 (1), 147–161.

Lim, S.J., Son, M., Ki, S.J., Suh, S.-I., Chung, J., 2023. Opportunities and challenges of machine learning in bioprocesses: categorization from different perspectives and future direction. Bioresour. Technol. 370, 128518.

Makowski, E.K., Chen, H.-T., Wang, T., Wu, L., Huang, J., Mock, M., Underhill, P., Pelegri-O'Day, E., Maglalang, E., Winters, D., et al., 2024. Reduction of monoclonal antibody viscosity using interpretable machine learning. In: Mabs. Vol. 16, (1), Taylor & Francis, 2303781.

Martínez, V.S., Buchsteiner, M., Gray, P., Nielsen, L.K., Quek, L.-E., 2015. Dynamic metabolic flux analysis using B-splines to study the effects of temperature shift on CHO cell metabolism. Metab. Eng. Commun. 2, 46–57.

Nik-Khorasani, A., Mehrizi, A., Sadoghi-Yazdi, H., 2024. Robust hybrid learning approach for adaptive neuro-fuzzy inference systems. Fuzzy Sets and Systems 108890.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., 2011. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.

Pérez-Rodriguez, S., Ramírez-Lira, M.d.J., Trujillo-Roldán, M.A., Valdez-Cruz, N.A., 2020. Nutrient supplementation strategy improves cell concentration and longevity, monoclonal antibody production and lactate metabolism of Chinese hamster ovary cells. Bioengineered 11 (1), 463–471.

Pramod, C., Tomar, M.S., Pillai, G.N., 2019. A modified extreme learning ANFIS for higher dimensional regression problems. In: Computational Intelligence: Theories, Applications and Future Directions-Volume II: ICCI-2017. Springer, pp. 279–292.

Rafiei, H., Akbarzadeh-T, M.-R., 2022. Reliable fuzzy neural networks for systems identification and control. IEEE Trans. Fuzzy Syst. 31 (7), 2251–2263.

Ritacco, F.V., Wu, Y., Khetan, A., 2018. Cell culture media for recombinant protein expression in Chinese hamster ovary (CHO) cells: History, key components, and optimization strategies. Biotechnol. Prog. 34 (6), 1407–1426.

Ruta, D., Gabrys, B., Lemke, C., 2011. A generic multilevel architecture for time series prediction. IEEE Trans. Knowl. Data Eng. 23 (3), 350–359. http://dx.doi.org/10.1109/tkde.2010.137.

Sankar, K., Hoi, K.H., Yin, Y., Ramachandran, P., Andersen, N., Hilderbrand, A., McDonald, P., Spiess, C., Zhang, Q., 2018. Prediction of methionine oxidation risk in monoclonal antibodies using a machine learning method. In: MAbs. vol. 10, (8), Taylor & Francis, pp. 1281–1290.

Schmitt, J., Razvi, A., Grapentin, C., 2023. Predictive modeling of concentration-dependent viscosity behavior of monoclonal antibody solutions using artificial neural networks. In: MAbs. vol. 15, (1), Taylor & Francis, 2169440.

Shah, P., Sheriff, M.Z., Bangi, M.S.F., Kravaris, C., Kwon, J.S.-I., Botre, C., Hirota, J., 2022. Deep neural network-based hybrid modeling and experimental validation for an industry-scale fermentation process: Identification of time-varying dependencies among parameters. Chem. Eng. J. 441, 135643.

Sheikholeslami, Z., Jolicoeur, M., Henry, O., 2014. Elucidating the effects of postinduction glutamine feeding on the growth and productivity of CHO cells. Biotechnol. Prog. 30 (3), 535–546.

Simpson, P.K., 1992. Fuzzy min-max neural networks. I. Classification. IEEE Trans. Neural Netw. 3 (5), 776–786.

Simpson, P.K., 1993. Fuzzy min-max neural networks-part 2: Clustering. IEEE Trans. Fuzzy Syst. 1 (1), 32.

Simpson, P.K., Jahns, G., 1993. Fuzzy min-max neural networks for function approximation. In: IEEE International Conference on Neural Networks. IEEE, pp. 1967–1972.

Sung, H.G., 2004. Gaussian Mixture Regression and Classification. Rice University.

Tagliaferri, R., Eleuteri, A., Meneganti, M., Barone, F., 2001. Fuzzy min–max neural networks: from classification to regression. Soft Comput. 5, 69–76.

Tsopanoglou, A., del Val, I.J., 2021. Moving towards an era of hybrid modelling: advantages and challenges of coupling mechanistic and data-driven models for upstream pharmaceutical bioprocesses. Curr. Opin. Chem. Eng. 32, 100691.

Waight, A.B., Prihoda, D., Shrestha, R., Metcalf, K., Bailly, M., Ancona, M., Widatalla, T., Rollins, Z., Cheng, A.C., Bitton, D.A., et al., 2023. A machine learning strategy for the identification of key in silico descriptors and prediction models for IgG monoclonal antibody developability properties. MAbs 15 (1), 2248671.

Walczak, B., Massart, D., 1996. The radial basis functions—partial least squares approach as a flexible non-linear regression technique. Anal. Chim. Acta 331 (3), 177–185.

Yu, H., Lu, J., Liu, A., Wang, B., Li, R., Zhang, G., 2022. Real-time prediction system of train carriage load based on multi-stream fuzzy learning. IEEE Trans. Intell. Transp. Syst. 23 (9), 15155–15165.

Zhang, D., Del Rio-Chanona, E.A., Petsagkourakis, P., Wagner, J., 2019. Hybrid physics-based and data-driven modeling for bioprocess online simulation and optimization. Biotechnol. Bioeng. 116 (11), 2919–2930.

Zhang, H., Sun, B., Peng, W., 2024. A novel hybrid deep fuzzy model based on gradient descent algorithm with application to time series forecasting. Expert Syst. Appl. 238, 121988.

Zhong, W., Fu, X., Zhang, M., 2022. A muscle synergy-driven anfis approach to predict continuous knee joint movement. IEEE Trans. Fuzzy Syst. 30 (6), 1553–1563.