NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTAMENT OF CHEMICAL ENGINEERING

Hybrid Deep Modeling of Biotechnological Processes: Combining Deep Neural Networks with First Principles Knowledge

José Miguel Machado de Matos Pinto

Master in Chemical and Biochemical Engineering

DOCTORATE IN CHEMICAL AND BIOLOGICAL ENGINEERING

NOVA University Lisbon

March, 2024

DEPARTAMENT OF
CHEMICAL ENGINEERING

Hybrid Deep Modeling of Biotechnological Processes: Combining Deep Neural Networks with First Principles Knowledge

José Miguel Machado de Matos Pinto
Master in Chemical and Biochemical Engineering

Adviser: Rui Manuel Freitas Oliveira,
Full Professor, NOVA School of Science and Technology

Co-adviser: Gerald Striedner
Full Professor, University of Natural Resources and Life Sciences, Vienna

Examination Committee:

Chair: José Paulo Barbosa Mota,
Full Professor, NOVA School of Science and Technology

Rapporteurs: Marco Paulo Seabra dos Reis,
Associate Professor with Habilitation, Faculty of Science and Technology of University of Coimbra

Fernando Gomes Martins,
Associate Professor, Faculty of Engineering of the University of Porto

Adviser: Rui Manuel Freitas Oliveira,
Full Professor, NOVA School of Science and Technology

Members: Pétia Georgieva,
Associate Professor with Habilitation, University of Aveiro

José Paulo Barbosa Mota,
Full Professor, NOVA School of Science and Technology

Maria D´Ascensão Carvalho Fernandes Miranda Reis,
Full Professor, NOVA School of Science and Technology

DOCTORATE IN CHEMICAL AND BIOLOGICAL ENGINEERING

NOVA University Lisbon
March, 2024

Hybrid Deep Modeling of Biotechnological Processes: Combining Deep Neural Networks with First Principles Knowledge

I dedicate this to my family, who have always been there for me

ACKNOWLEDGMENTS

I want to thank my advisers, Professor Rui Oliveira and Professor Gerald Striedner for having offered me this chance and for the guidance they offered every step of the way.

To Professor Rui Oliveira, I want to offer special thanks for giving me the chance to achieve this milestone, for all the many ideas and experience he shared and for always being available to hear my ideas, even the most far-fetched ones. You helped me grow as a researcher far beyond anything I could think of.

I want to thank FCT – Fundação para a Ciência e a Tecnologia for the scholarship grant that allowed me to carry out this research, NOVA School of Science and Technology for giving its students all the conditions needed to flourish and the University of Natural Resources and Life Sciences for the data they acquired.

I also want to thank everyone in the lab. To João Ramos, for the many times we worked together and the meaningful discussions that came from it. To Rafael Costa, for the immeasurable help when dealing with Python and the SBML2HYB tool. To Rosa, for agreeing to test out my code and her cheerfulness. To Monesh, for being so dedicated as he starts his path into hybrid modeling. And to all the many students who passed through this lab, I'm sorry that I can't name you all, but I thank you for the many ideas you brainstormed with me.

I want to thank my friends for the help they gave, especially to Sofia. As a fellow pursuer of a doctoral degree, she knows things don't always go as we want, but her work ethic always pushes me to work harder. If it weren't for our banter and her wildly off-topic suggestions this thesis wouldn't have taken this shape.

Lastly, I want to thank my family for the boundless support they gave me. I want to thank my parents, António and Teresa, for helping me get this far. I know I have always been a complicated person, but your love and patience are always far more than anyone could hope for. I also want to thank my sister, Laura, for being a source of strength. Knowing that you look up

to me gives me the motivation to push through no matter how complicated things may be. I also want to thank my aunt, Ilda, my uncle, Ricardo, and my cousin Inês, for always showing interest in my work, for the support they give me and for all the pep talks.

I also want to dedicate this thesis and give the most profound thanks to those who unfortunately can't be here to see this but were critical for my journey to be a success. To my maternal grandmother Bela, who was like a second mother to me, who celebrated all my victories and who always had a smile ready for me. To my maternal great-grandmother, who so often watched over me as a small child. To my maternal grandfather, António, for always pushing me to go further in my studies and believing in my potential. To my paternal grandmother Carminda, for reminding me of the importance of family every time. To my paternal grandfather, António, for all the pride he showered me with and the support.

And, of course, I also want to dedicate this thesis to the memory of my maternal great-grandfather and my namesake, José. I may have never met him, but from all the stories I heard I wish I had. He was always a source of inspiration for the joy and the smiles that he put on the faces of those who spoke about him, and I hope to one day be able to do the same.

"You cannot teach a man anything; you can only help him discover it in himself." (Galileo).

ABSTRACT

Hybrid modeling combining First-Principles with Machine Learning (ML) is becoming a pivotal methodology for Biopharma 4.0 enactment. Combining ML with prior knowledge generally improves the model predictive power and model transparency while reducing the amount of data for process development. However, most previous studies pursued a shallow hybrid modeling approach based on three-layered Feedforward Neural Networks (FFNNs) combined with macroscopic material balance equations.

In this thesis, a general deep hybrid modelling framework for bioreactors, that incorporates deep neural networks, deep learning and First Principles equations is developed and implemented in the HYBrid MODdeling (HYBMOD) MATLAB® toolbox (Chapter 3). Deep learning, namely the adaptive moment estimation method (ADAM), stochastic regularization and depth-dependent weights initialization are evaluated in a hybrid modeling context. Modified sensitivity equations are proposed for the computation of gradients in order to reduce CPU time for the training of deep hybrid models. Furthermore, the encoding of hybrid models obeying to the Systems Biology Markup Language (SBML) standard is implemented.

The general deep hybrid modeling method is evaluated in several experiments using synthetic and real-world experimental data. In Chapter 4 a pilot *Pichia pastoris* GS115 MUT+ process development case study is addressed. In Chapter 5 an industrial CHO-K1 process development campaign is addressed. The results point to the conclusion that there is a clear advantage of deep hybrid modeling both in terms of predictive power and in terms of computational cost in relation to the shallow hybrid case. Furthermore, the SBML compatibility facilitates the dissemination of hybrid models in the Systems Biology community.

Keywords: Hybrid modeling, FFNNs, Deep learning, ADAM, SBML, Bioreactors, *Pichia pastoris*, CHO-K1, Biopharma 4.0.

RESUMO

Modelação híbrida que combina Primeiros-Princípios com Aprendizagem de Máquinas (ML) está a tornar-se numa metodologia fundamental da *Biopharma* 4.0. Como vantagens aponta-se a melhoria do poder preditivo do modelo bem como a sua transparência, reduzindo ainda a quantidade de dados para o seu desenvolvimento. No entanto, a maioria dos estudos anteriores seguiu uma abordagem de modelação híbrida não-profunda baseada em Redes Neuronais (FFNNs) de três camadas e equações de balanços materiais macroscópicos.

Nesta tese, modelação híbrida profunda de biorreatores que incorpora redes neurais, aprendizagem profunda e Primeiros-Princípios é desenvolvida e implementada na *toolbox* HYBrid MODdeling (HYBMOD) em MATLAB® (Capítulo 3). A aprendizagem profunda, nomeadamente o método adaptativo de estimação de momento (ADAM), regularização estocástica e inicialização de pesos dependentes da profundidade são avaliadas. Equações de sensibilidade modificadas são propostas para o cálculo de gradientes, a fim de reduzir o tempo de CPU para o treino de modelos híbridos profundos. Além disso, é implementada a codificação de modelos híbridos obedecendo ao padrão SBML (*Systems Biology Markup Language*).

O método proposto de modelação híbrida profunda é avaliado usando dados experimentais sintéticos e do mundo real. No Capítulo 4 é abordado o desenvolvimento de processos *Pichia pastoris* GS115 MUT+ à escala piloto. No Capítulo 5 é estudado o desenvolvimento duma cultura industrial de CHO-K1. Os resultados apontam para uma clara vantagem da modelação híbrida profunda tanto em termos de poder preditivo quanto em termos de custo computacional em relação à modelação híbrida não-profunda. Além disso, a compatibilidade SBML facilita a disseminação de modelos híbridos na comunidade de Biologia de Sistemas.

Palavras chave: Modelação híbrida, FFNNs, Aprendizagem profunda, ADAM, SBML, Biorreatores, *Pichia pastoris*, CHO-K1, *Biopharma* 4.0.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

ANN	Artificial Neural Network
FFNN	Feed Forward Neural Network
SBML	Systems Biology Markup Language
CPP	Critical Process Parameter
ADAM	Adaptive Moment Estimation Method
ScFv	Single-chain Variable Fragment
M3C	Measurement, Modeling, Monitoring and Control
ODE	Ordinary Differential Equation
CSTR	Constant Stir Type Reactor
PBE	Population Balance Equation
CHO	Chinese Hamster Ovary
GEM	Genome-scale reconstructed Model
HEK	Human Embryo Kidney
FBA	Flux Balance Equation
GMA	Generalized Mass Action
ML	Machine Learning
DR	Dimensionality Reduction
RL	Reinforcement Learning
ReLU	Regularized Linear Unit
PINN	Physics Informed Neural Network
MLP	Multilayer Perceptron
tanh	Hyperbolic tangent
DT	Decision Tree
RF	Random Forest

IVP	Initial Value Problem
CPU	Central Processing Unit
RAM	Random Access Memory
CCD	Central Composite Design
DoE	Design of Experiment
AICc	Akaike Information Criterion with second order correction
MUT+	Methanol Utilization Plus
GB	Glycerol Batch
GFB	Glycerol Fed-batch
MFB	Methanol Fed-batch
LMM	Levenberg-Marquardt Method
CV	Cross Validation
WSSE	Weighted Sum of Square Error
PLC	Recombinant Phospholipase C
BSM	Basal Salt Medium
PID	Proportional Integral Derivative
PCA	Principal Component Analysis
WMSE	Weighted Mean Square Error
CC-DOE	Central Composite Design of Experiments

SYMBOLS

$\boldsymbol{x(t)}$	State vector
$\boldsymbol{u(t)}$	Control vector
$\boldsymbol{y(t)}$	Measured variables vector
θ_x	State-space parameters vector
θ_y	Measurement model parameters vector
c	Vector of concentrations of extracellular compounds
r	Vector of volumetric reaction rates
D	Dilution rate
F	Volumetric feeding rate
V	Liquid volume inside the reactor
c_{in}	Concentrations of extracellular compounds in the inlet stream
q	Vector of volumetric exchange rates from gas to liquid phase
X	Biomass
S	Substrate
P	Product
SH	Memory shock factor
μ	Specific growth rate
m_P	Specific rate of product formation

m_S	Maintenance rate
$y_{i/j}$	Stoichiometric coefficient
μ_{max}	Maximum cell growth rate
$K_{I,s}$	Substrate inhibition constant
I	Inhibitor
v_P	Specific product synthesis rate
$w(y,t)$	Distribution function
$r(y,t)$	Rate of change of property y
$h(y,t)$	Net rate of formation of cells with property y
z	Vector of concentrations of intracellular species
v	Vector of specific reaction rates
T	Temperature
pH	pH
θ	Kinetic parameters
w	Network parameters
b	Network bias
σ	Standard deviation
α	ADAM learning rate
β_1	ADAM update rate 1
β_2	ADAM update rate 2
ε	ADAM minimum value
nw	Number of network + bias weights

1

INTRODUCTION

1.1 Thesis Motivation

Hybrid modelling is becoming a pivotal methodology for process digitalization in the chemical and biological industries. The combination of artificial neural networks with physical laws in hybrid model structures has proven to be advantageous in a large number of case studies described in the literature. The main motivation was to overcome neural networks limitations, namely the i) inability to comply with process constraints, ii) the tendency for data overfitting, and iii) the poor predictive power outside the training-validation domain. Hybrid modelling workflows combining neural networks with physical laws enable a more rational usage of prior knowledge eventually translating into more accurate, transparent and robust process models. Most of previous hybrid modelling studies combined shallow neural networks with physical laws. Recent advances in deep learning have however demonstrated that neural networks with multiple hidden layers (deep networks) are advantageous over their shallow counterparts. Only very recently the deep learning advances are penetrating the hybrid modeling field. There is a considerable research gap concerning the use of deep neural networks and deep learning algorithms such as the Adaptive Moment Estimation Method (ADAM) and stochastic regularization in the hybrid modelling field. The present PhD dissertation addresses this research gap.

1.2 Objectives

The main objectives of this work are the development of a deep hybrid modelling method that combines mechanistic models with emergent deep neural networks and the implemen-

tation of these developed hybrid modelling methods in a way that is scalable to large Systems Biology models and Systems Biology Markup Language (SBML) compatible. More specifically, the following objectives are targeted:

- Develop deep hybrid modelling structures combining mechanistic knowledge with emergent deep neural networks.
- Develop efficient training algorithms for deep hybrid models that are stable and cost efficient in terms of computation time.
- Develop a methodology to encode hybrid model structures obeying to the SBML standard.
- Implement the deep hybrid modelling methodology in the HYBMOD MATLAB/Python toolbox.
- Benchmark shallow hybrid modelling and deep hybrid modelling in several case studies.
- Showcase the deep hybrid modelling framework with real life cultivation processes.

1.3 Thesis Organization

This thesis is divided in 7 chapters:

Chapter 1 is an introductory chapter where the main objectives and contributions of the thesis, along with its organization, are laid out for simplicity.

Chapter 2 focuses on the topic of dynamic modeling for bioreactor monitoring, optimization, and control applications. The first part of the chapter overviews mechanistic modeling across different scales, covering the concepts of structured/unstructured, segregated/unsegregated and genome-scale modeling. The second part of the chapter covers machine learning methods for supervised, unsupervised and reinforced learning in a bioprocessing context, with emphasis on building supervised bioreactor models that improve with process experience. Knowledge abstraction in the machine learning world is hardly compatible with the vast wealth of engineering and scientific knowledge accumulated over decades in the form of mechanistic models. The opportunities to develop hybrid mechanistic/machine learning models for bioreactors in the context of Industry 4.0 are finally highlighted. The vision is that machine learning should augment mechanistic bioreactor models rather than replace them.

Chapter 3 revisits the general bioreactor hybrid model and introduces deep learning techniques. Multi-layer networks with varying depths were combined with First Principles equations in the form of deep hybrid models. Deep learning techniques, namely the adaptive

moment estimation method (ADAM), stochastic regularization and depth-dependent weights initialization were evaluated in a hybrid modeling context. Modified sensitivity equations are proposed for the computation of gradients in order to reduce CPU time for the training of deep hybrid models. The methods are illustrated with applications to a synthetic dataset and a pilot 50 L MUT+ *Pichia pastoris* process expressing a single chain antibody fragment.

In Chapter 4, a hybrid deep modeling method with state-space reduction is developed and showcased with a *P. pastoris* GS115 Mut+ process expressing a single-chain antibody fragment (scFv). Deep feedforward neural networks (FFNN) with varying depths were connected in series with bioreactor macroscopic material balance equations. The hybrid model structure was trained with a deep learning technique based on the adaptive moment estimation method (ADAM), semidirect sensitivity equations and stochastic regularization. A state-space reduction method was investigated based on principal component analysis (PCA) of cumulative reacted amount. Data of nine fed-batch *P. pastoris* 50 L cultivations served to validate the method. Hybrid deep models were developed describing process dynamics as function of critical process parameters (CPPs).

Chapter 5 compares, for the first time, deep and shallow hybrid modeling in a CHO process development context. Data of 24 fed-batch cultivations of a CHO-K1 cell line expressing a target glycoprotein, comprising 30 measured state variables over time, were used to compare both methodologies. Hybrid models with varying FFNN depths (3-5 layers) were systematically compared using two training methodologies. The classical training is based on the Levenberg-Marquardt algorithm, indirect sensitivity equations and cross-validation. The deep learning is based on the Adaptive Moment Estimation Method (ADAM), stochastic regularization and semidirect sensitivity equations.

In Chapter 6, a computational framework is proposed that merges mechanistic modeling with deep neural networks obeying the Systems Biology Markup Language (SBML) standard. With the proposed framework, existing SBML models may be redesigned into hybrid systems through the incorporation of deep neural networks into the model core, using a freely available python tool. The so-formed hybrid mechanistic/neural network models are trained with a deep learning algorithm based on the adaptive moment estimation method (ADAM), stochastic regularization and semi-direct sensitivity equations. The trained hybrid models are encoded in SBML and uploaded in model databases, where they may be further analyzed as regular SBML models. This approach is illustrated with three well-known case studies: the Escherichia coli threonine synthesis model, the P58IPK signal transduction model, and the Yeast glycolytic oscillations model.

Lastly, Chapter 7 presents the main conclusions obtained from Chapters 3 to 6 and proposes areas where there may be future applications of hybrid models.

1.4 Thesis Contributions

The main contributions of this thesis are twofold.

First, a novel deep hybrid model methodology that takes advantage of the deep learning paradigm is created. This approach is shown to have a higher degree of fidelity to the real process when compared to the standard non-deep hybrid approach, as showcased with both synthetic and real-world experiments.

Second, the development of this methodology is done in such a way that new models can be quickly developed while maintaining a common organization amongst them. This type of systematic approach to modelling allows them to be compatible with the SBML format, facilitating their analysis with Systems Biology tools.

The results of the research activities of this PhD dissertation originated the following five publications in peer-reviewed journals and one chapter book:

- Pinto, J., Antunes, J., Ramos, J., Costa, R. S., & Oliveira, R. (2022). Modeling and optimization of bioreactor processes. In *Current Developments in Biotechnology and Bioengineering* (pp. 89-115). Elsevier.
- Pinto, J., Mestre, M., Ramos, J., Costa, R. S., Striedner, G., & Oliveira, R. (2022). A general deep hybrid model for bioreactor systems: Combining first principles with deep neural networks. *Computers & Chemical Engineering*, *165*, 107952.
- Pinto, J., Ramos, J. R., Costa, R. S., & Oliveira, R. (2023). Hybrid Deep Modeling of a GS115 (Mut+) Pichia pastoris Culture with State–Space Reduction. *Fermentation*, *9*(7), 643.
- Pinto, J., Costa, R. S., Alexandre, L., Ramos, J., & Oliveira, R. (2023). SBML2HYB: a Python interface for SBML compatible hybrid modeling. *Bioinformatics*, *39*(1), btad044.
- Pinto, J., Ramos, J. R., Costa, R. S., & Oliveira, R. (2023). A General Hybrid Modeling Framework for Systems Biology Applications: Combining Mechanistic Knowledge with Deep Neural Networks under the SBML Standard. *AI*, *4*(1), 303-318.
- Pinto, J., Ramos, J. R., Costa, R. S., Rossell, S., Dumas, P., & Oliveira, R. (2023). Hybrid deep modeling of a CHO-K1 fed-batch process: combining first-principles with deep neural networks. *Frontiers in Bioengineering and Biotechnology*, *11*.

2

State of the Art

2.1 Introduction

Mathematical modeling is in its essence the translation of prior knowledge regarding the system at study into a compact mathematical representation. The translation of knowledge into a mathematical construct can be performed in many different ways, resorting to many different mathematical formalisms. This chapter focuses on a particular class of bioreactor dynamic models for Measurement, Modeling, Monitoring and Control (M3C) applications (Mandenius, 2004, Carrondo et al., 2012). Bioreactor dynamic models for M3C are essential tools to speed-up bioprocess development or for the control of large-scale production bioreactors. The aim of M3C models is to establish a quantitative cause-effect relationship between control degrees of freedom, state variables, measured variables, and a profit function, which is dynamic in nature. Such models are used for off-line simulation and optimization (the open-loop dynamic optimization problem), for on-line state and/or parameters estimation, for model predictive control, among many other applications. Recently, bioreactor dynamic models are being considered for the implementation of digital twins in the context of Industry 4.0 (Nargund and Mauch, 2019, McLamore et al, 2020, Moser et al., 2020, Jens et al, 2020).

Bioreactors are complex multi-scale processes that are very challenging to model (Figure 1). For dynamic modeling of stirred tank bioreactors, homogeneity of the macroscopic scale is normally assumed. In scale-up problems, the understanding of the macroscopic heterogeneity becomes essential, for which the development of computational fluid dynamics models becomes a major challenge (not covered in this chapter). Cell cultures are in reality comprised of heterogeneous mixtures of cells that differ with regard to size, mass and intracellular concentrations of proteins, DNA and other chemical constituents. In many problems, population heterogeneity is an important factor to consider thereby substantially increasing the complexity of the model (Ataai and Shuler 1985; Domach and Shuler 1984; Henson 2003a; Sidoli et al. 2004). The intracellular processes comprehend thousands of metabolic reactions and many more regulatory processes involving genes, RNAs, proteins and metabolites. In the last two decades, systems biology has led to an explosion of knowledge regarding intracellular processes, that can now be integrated in bioreactor models. For some bioreactor dynamic modeling problems there is no need to consider all the scales with a high level of detail. Multi-scale modeling becomes however critical when the product quality attributes are expressed at the molecular level (e.g. Glycosylation quality attributes of a biologic), wherefore all the

scales (molecular-micro-macro) potentially play a role, with the complexity of the model dramatically exploding.

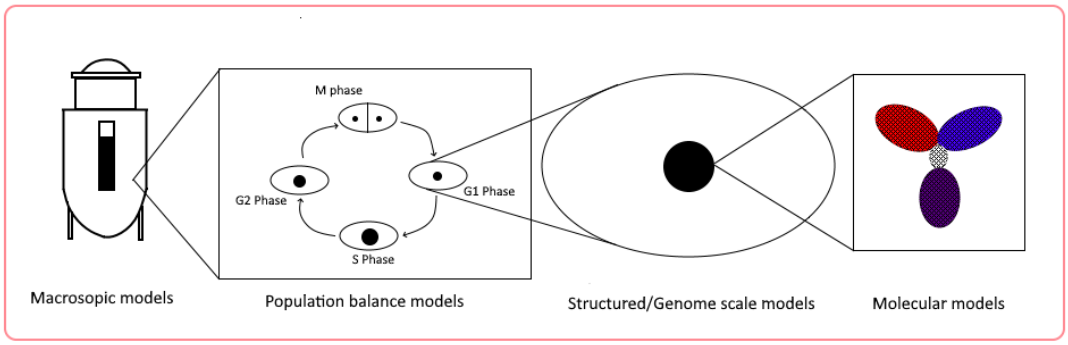

Figure 1. The multiscale nature of a bioreactor system

There are currently two apparently conflicting approaches to address such complex bioreactor modeling problems (Baker et al., 2018): mechanistic modeling and data-based/machine learning. From one side, mechanistic modeling based on First Principles of physics, chemistry and biology has been the classical approach to develop bioreactor models. First Principles include the conservation laws of mass, energy and momentum, which may be stated for a bioreactor *ab initio* without the need of experimental evidence. Mechanistic models are frequently complemented with phenomenological and/or semi-empirical models to describe less defined parts of the process, for which prior mechanistic knowledge is still missing.

The other emerging vision is that of machine learning leveraged on high throughput data sets across different scales. Technological advances in bioprocesses research have pushed high-throughput instruments, with increasingly accurate data being collected (particularly proteomics, metabolomics, transcriptomics, and genomics data) (Palsson 2002). With Industry 4.0 enactment, such measurement devices will become widespread and deliver large-scale collections of datasets from heterogeneous sources, called in computational science as "big data" (Cook et al., 2018). Significant effort has been put into ensuring the scalability of computational tools for the collection of these massive bioprocess data, but analysis and integration remains a challenge (Qin et al., 2015). The availability of data has been one of the most notable advances in predictive modeling. With this background, the deployment of machine learning in a bioprocessing context will likely grow in the future, including the application for bioreactor modeling, optimization and control (the M3C challenge). Particularly, machine learning offers the possibility of modeling complex bioreactor data sets across multiple scales, with the ability to identify patterns and learn and improve through time, thereby real-

izing Industry 4.0 vision (Jordan and Mitchell 2015). In parallel to the emergence of machine learning, a new movement towards hybrid approaches that combine mechanistic modeling with machine learning is getting momentum (Galvanauskas et al., 2004, Oliveira, 2004, von Stosch et al., 2014). The vision is that machine learning should be used to augment mechanistic models rather than to replace them. Hybrid modeling combines the power of mechanistic understanding and predictive modeling thus particularly attractive for tackling complex bioreactor modeling problems.

The first part of this chapter overviews the key mechanistic modeling concepts for a bioreactor system emphasizing the multi-scale nature and the many challenges yet to overcome. In particular, the reduction and integration of genome-scale models with bioreactor models is discussed. The second part of the chapter covers machine learning methods for supervised, unsupervised and reinforced learning in a bioprocessing context, with emphasis on building supervised bioreactor models that improve with process experience. It finalizes with an overview of hybrid mechanistic/machine learning models for bioreactors in the context of Industry 4.0.

2.2 The Traditional Approach: Bioreactor Mechanistic Models

A bioreactor mathematical model may be expressed in different ways depending on the objective of the model. This chapter will address a particular class of dynamic models for perfectly mixed bioreactors expressed by the following general state-space representation (Equation 1 and Equation 2):

Equation 1. State space model of a perfectly mixed bioreactor

$$\frac{dx}{dt} = f(x(t), u(t), \theta_x, t); \quad x(t_0) = x_0$$

Equation 2. Measurement model of a perfectly mixed bioreactor

$$y = h\big(x(t), \theta_y\big)$$

with x(t) the state vector, $u(t)$is the control vector, $y(t)$the vector of measured variables, θ_x and $\theta_y$are the parameters vector of the state–space and measurement models respectively, and t the dependent variable time. Equation 1 is the state-space model while Equation 2 is the measurement model. The functions $f(\cdot)$ and $h(\cdot)$expressed by Equation 1 and Equation 2 are typically complex and nonlinear, which render bioreactors rather complex dynamical sys-

tems, which are difficult to identify and control. The system of Equation 1 and Equation 2 may be shaped in many different ways depending on the level of detail of the knowledge available, as shown in the proceeding sections.

2.2.1 Macroscopic material balances

Macroscopic bioreactor dynamics may be established *ab initio* by the material balance equations of the key extracellular compounds that intervene in the reaction mechanism. These material balances are expressed by systems of ordinary differential equations (ODEs), which take the following general state-space form (Bastin and Dochain, 1990):

Equation 3. General state-space equation

$$\frac{dc}{dt} = r - Dc + Dc_{in} + q$$

where c is a vector of n concentrations of extracellular compounds (the state vector), r is a vector of n volumetric reaction rates, $D = \frac{F}{V}$ is the dilution rate (F is the volumetric feeding rate into the reactor and V the liquid volume inside the reactor), c_{in} is a vector of n concentrations of extracellular compounds in the inlet stream, and $q$is a vector of n volumetric exchange rates from the gas to the liquid phase, which apply to gases (e.g. O_2, CO_2, H_2, CH_4, etc.…) and extracellular volatile compounds which are typically low molecular weight metabolites resulting from the central carbon metabolism (e.g., ethanol, methanol, etc.…). Equation 3 must be complemented with the general mass balance equation. If the specific mass of the inlet stream is not significantly different than that of the liquid inside the bioreactor, the following simplified general mass balance applies:

Equation 4. General mass balance equation

$$\frac{dV}{dt} = DV$$

Equation 3 and Equation 4 are generic for perfectly mixed bioreactors irrespective of the operation mode. In batch reactors, Equation 3 and Equation 4 hold with $D = 0$. Fed-batch bioreactors are expressed by system of Equation 3 and Equation 4 without modifications. A CSTR in transient operation is modeled with Equation 3 plus $\frac{dV}{dt} = F - F = 0$ (volume is constant). A CSTR in steady state is modeled with system Equation 3 and Equation 4 by making all derivatives equal null, i.e. $0 = r - Dc + Dc_{in} + q$ and $\frac{dV}{dt} = 0$.

The system of Equation 3 and Equation 4 must be completed with defining kinetic equations for the gas-liquid volumetric transfer rate, q, and for the volumetric reaction term, r. Gas-liquid mass transfer models are well covered in reference textbooks (e.g. Bailey and Ollis, 1986, Blanch and Clarck, 1996) and will not be covered here. The critical challenge in bioreactor engineering is the modeling of the reaction kinetics, which will be further covered below.

2.2.2 Unstructured growth models

The simplest approach to define a bioreactor model is by considering the cells all equal (unsegregated) and without intracellular structure (unstructured). These assumptions give rise to unstructured growth models, which were the prevailing type of models until the early 00s. The extracellular compounds are considered as biochemical species that intervene in a simplified bio-reaction mechanism, with biomass the catalyst of such bio-reactions. As illustrative example, a simple bio-reaction mechanism whereby biomass (X) grows on a limiting substrate (S) resulting in the formation of biomass itself (X) and product (P), may be expressed as (Equation 5):

Equation 5. Examples of an unstructured growth model

$$y_{S/X}S \xrightarrow{\mu} 1\,X + y_{P/X}P$$

$$y_{S/P}\,S \xrightarrow{m_P} 1\,P$$

$$S \xrightarrow{m_S} \emptyset$$

$$X \xrightarrow{k_d} \emptyset$$

The first reaction represents cell growth with specific growth rate μ and with concomitant formation of product (P) (cell growth associated product synthesis). The second reaction represents cell growth dissociated product formation with specific rate of product formation m_P. The third reaction represents the substrate metabolized for cellular maintenance with maintenance rate, m_S (the symbol $\emptyset$ represents an unspecified entity/species). The fourth reaction represents biomass death with death rate k_d. The yields $y_{i/j}$ are stoichiometric coefficients typically defined on a mass basis due to the undefined nature of some biochemical species, particularly biomass. The specific growth rate is normally defined by the Monod model (Monod, 1949) to express growth limitation by substrate S (Equation 6):

Equation 6. Monod model for biomass growth

$$\mu = \mu_{max} \frac{[S]}{K_S + [S]}$$

where $\mu_{max}$is the maximum cell growth rate, $[S]$ is the substrate concentration, and K_S is the Monod constant. The Monod model is inspired in the irreversible Michaelis-Menten enzyme kinetics but is of empirical nature. To express growth inhibition by high substrate concentrations, the Andrews model (Andrews, 1968) is a common choice (Equation 7):

Equation 7. Andrews model for biomass growth with high substrate inhibition

$$\mu = \mu_{max} \frac{[S]}{K_S + [S] + \frac{[S]^2}{K_{I,s}}}$$

where $K_{I,s}$is the substrate inhibition constant. The Andrews model is inspired by the Hans and Levenspiele acid-base equilibrium model for enzyme kinetics. Han and Levenspiel (1988) have extended the Monod model by considering the effect of $i = 1, \dots, h$ inhibitors of cell growth (Equation 8):

Equation 8. Han and Levenspiel model for biomass growth with multiple inhibitors

$$\mu = \mu_{max} \prod_{i=1}^{h} \left(1 - \frac{[I_i]}{[I_i^*]}\right)^{n_i} \left(\frac{[S]}{[S] + K_S \prod_{i=1}^{h} \left(1 - \frac{[I_i]}{[I_i^*]}\right)^{m_i}}\right)$$

where $[I_i]$ is the concentration of inhibitor I_i. This model assumes the existence of a critical inhibitor concentration $[I_i^*]$ above which cells cannot grow, and that the constants of the Monod equation are functions of this limiting inhibitor concentration. The n_i andm_i are additional kinetic parameters that need to be estimated from data. The Monod model may also be extended to express the limitation of multiple n_s substrates and the inhibition of multiple n_p products (Equation 9):

Equation 9. Extended Monod model for multiple substrate limitation and product inhibitions

$$\mu = \mu_{max} \prod_{i=1}^{n_s} \frac{[S_i]}{K_{Si} + [S_i]} \prod_{i=1}^{n_P} \frac{K_{Pi}}{K_{Pi} + [P_i]}$$

The specific product synthesis rate, v_P, may be expressed by the Luedeking-Piret equation (Luedeking and Piret, 1959) (Equation 10):

Equation 10. Luedeking and Piret model for product growth

$$v_P = y_{P/X}\mu + m_P$$

which considers a growth associated product synthesis term and a growth dissociated term. These reaction kinetics and many other are simplified phenomenological models commonly referred to as Monod-type kinetics.

Linking with the macroscopic balance (Equation 3) needs the definition of the volumetric reaction rates, which for the case of unstructured growth models take the following general form (Equation 11):

Equation 11. Volumetric reaction rates general equation for unstructured growth models

$$r = K\, v(c, u, \theta)\, X$$

where $K = \{y_{i/j}\}$ is a $n \times m$ matrix of yield coefficients and $v(c, u, \theta)$is the vector of specific reaction rates (n is the number of species and m the number of reactions). The yield coefficients in matrix K have been often observed dependent of the experimental conditions. This apparent time-varying nature of yield coefficients is explained by the lack of intracellular structure in the model. Metabolic pathway analysis of genome scale networks has highlighted the redundant nature of biochemical networks, characterized by millions of metabolic circuits between extracellular substrates and end products in prokaryotes and even more in eukaryote cells. The ability to dynamically adjust intracellular states explains the time-varying nature of yield coefficients and the lack of predictive power of purely macroscopic models. In many cases the consideration of intracellular structure becomes mandatory, which will be further covered in the next sections.

2.2.3 Segregated growth models

Cell cultures are in reality comprised of heterogeneous mixtures of cells that differ with regard to size, mass and intracellular concentrations of proteins, DNA and other chemical constituents. To account for population heterogeneity, population material balance equations are applied to segregate groups of cells with identical properties. In a population balance based on cell number, the cells are differentiated in terms of a given set of properties, y. The distribution of cells in the population in relation to properties y is given by a distribution function $w(y, t)$, where $w(y, t)dy$ represents the number of cells per unit volume within the property interval$[y, y + dy]$ at time t. The total cell count is then given by(Equation 12):

Equation 12. Cell distribution function

$$w(t) = \int_{y_{min}}^{y_{max}} w(y, t)dy$$

The distribution function $w(y,t)$is obtained by solving the population balance equation (PBE), which for a homogeneous bioreactor without cell feeding, takes the following general form (Nielsen and Villadsen, 1994) (Equation 13):

Equation 13. General population balance equation

$$\frac{\partial w(y,t)}{\partial t} + \frac{\partial [r(y,t)w(y,t)]}{\partial y} = h(y,t) - Dw(y,t)$$

where $r(y,t)$ is the rate of change of property y, and $h(y,t)$ is the net rate of formation of cells with the property y due to cell division, and $D$is the dilution rate in the bioreactor. The net rate of formation of cells with the property y due to cell division maybe further detailed by splitting into the rates of formation and disappearance of cells with property y (Equation 14):

Equation 14. Net rate of cell formation

$$h(y,t) = 2\int_{y_{min}}^{ymax} (y',t)p(y,y',t)w(y',t)dy' - (y,t)w(y,t)$$

with (y',t) the division rate of cells with propertyy', $p(y,y',t)$ the probability of a mother cell with property y' dividing into 2 daughter cells with property y.

The link with the macroscopic material balances requires a modification of Equation 3 to account for the influence of properties y in the specific reaction rates, v, as follows (Equation 15):

Equation 15. General volumetric rates for segregated growth models

$$r = K\int_{y_{min}}^{y_{max}} v(c,y,u,\theta)\, w(y,t)dy$$

For simplicity, PBEs are usually applied to a single property such as cell age (A. Hjortso and Nielsen 1995) or cell mass (Nishimura and Bailey 1981). Since it considers growth and division of single cells, this approach can be used to describe heterogeneity caused by extra and intracellular fluctuations (Delvigne et al. 2014). Many PBE models have been developed (Anderson et al. 1969; Fadda et al. 2012; Ganusov et al. 2000; Henson 2003a; Zhu et al. 2000) and several numerical methods were developed to reduce the computational hurdles for solving the resulting nonlinear integro-ordinary differential and integro-partial differential equations (Liu et al. 1997; Mantzaris et al. 2001; Pigou et al. 2017; Singh et al. 2020).

2.2.4 Intracellular structure

Structured models further consider intracellular compartments (cytosol, mitochondria, nucleus, etc.…) and the concentrations of intracellular species (metabolites, proteins, RNA/DNA and other chemical constituents), which are involved in a very complex network of physiochemical transformations. Under the assumption of well-mixed compartments, the dynamic material balance equations over intracellular species are generically stated as Equation 16:

Equation 16. Intracellular material balance equations' general form

$$\frac{dz}{dt} = N^{\{z\}}v - \mu z$$

where z is a vector of concentrations of nz intracellular species, $N^{\{z\}}$is a $nz \times q$ stoichiometric matrix of intracellular reactions, v is a vector of q specific reaction rates (including transport reactions across compartments), and $\mu$is the specific growth rate. The second term in the right-hand side of Equation 16 expresses the dilution of intracellular species due to the increase of cell mass. The reaction rates, v, are complex functions of extracellular concentrations, c, intracellular concentrations, z, input variables (such as T, pH, etc) and kinetic parameters, θ (Equation 17):

Equation 17. General volumetric rates for segregated growth models

$$v = f(z, c, u, \theta)$$

The development of structured models has been historically limited by the lack of knowledge of the very complex intracellular phenomena. With the emergence of systems biology in the early 00s, several GEnome-scale reconstructed Models (GEM) have been developed for industrially relevant cells such as Escherichia coli (Monk et al., 2013), Saccharomyces cerevisiae (Foster et al., 2003), Pichia pastoris (Sohn et al., 2010), CHO cells (Hefzi et al., 2016), HEK cells (Quek et al., 2014) and many other. This new generation of GEMs are now being considered for bioreactor dynamic modeling, control and optimization. Construction of large dynamic GEMs has been attempted in two ways: i) traditional kinetic modeling paradigm, or ii) dynamic flux balance analysis (dynamic FBA) techniques (Stanford et al., 2013). Dynamic FBA avoids the definition of the kinetic rate Equation 17 by dynamic optimization of a cellular objective function (Mahadevan et al., 2002). However, rigorous dynamic modeling requires the definition of the kinetic Equation 17. One advantage of GEMs is the association between genes, enzymes, reactions and respective catalytic mechanisms. For bi-molecular metabolic reac-

tions the mechanistic Michaelis-Menten model generally applies (Cornish-Bowden, 1995). A large number of metabolic reactions involve more than 1 substrate or more than 1 product. For such reactions, Liebermeister et al. (2010) proposed a generalized form of the reversible Michaelis-Menten model called modular rate law, which is now popular for GEM. Unfortunately, most of the reaction mechanisms in GEMs are unknown and approximations are required, such as generalized mass action (GMA), hill kinetics, lin-log kinetics (Visser and Heijnen, 2003), convenience kinetics (Liebermeister and Klipp, 2006) and power laws (Savageau, 1970) and their combinations (Costa et al., 2010). GMAs are simplistic approximations of the reaction mechanisms based on the principle that the reaction rate is proportional to the probability of collision of reactant molecules. GMAs have only 2 parameters and they can be automatically generated from the reaction stoichiometry, which have popularized them in GEM.

The link with the macroscopic material balances Equation 3 is not explicit in Equation 17. A subset of reactions in vector, v, is associated with transport processes across the cellular membrane for the exchange between intracellular and extracellular compartments. The net volumetric reaction rate of n extracellular species can then be calculated as Equation 18:

Equation 18. Volumetric reaction rates from an intracellular structure model

$$r = X\, N^{\{c\}}\, v$$

where $N^{\{c\}}$ is the $n \times q$ stoichiometric matrix associated with n extracellular compounds. Equation 18 links with the macroscopic material balances Equation 3, which may be rewritten as Equation 19:

Equation 19. General state-space equations for extracellular components when using an intracellular model.

$$\frac{dc}{dt} = X\, N^{\{c\}}\, v - Dc + q$$

Given the very large size of $N^{\{z\}}$, $N^{\{c\}}$ and v with thousands of species and reactions, it is critical to reduce GEMs to the reactor operating conditions. As illustrative example, Quek et al. (2014) have adapted the RECON-2 model (Thiele et al., 2013) with 7440 reactions for the cultivation of HEK293 cells in a defined medium, with a significative reduction to 329 reactions.

2.3 Bioreactor Models for Industry 4.0

Industry 4.0 is now widely accepted as the next paradigm for production with the widespread of automation, data connectivity and machine learning (ML). ML is an essential part of Industry 4.0 that allows systems and algorithms to automatically improve based on experience. This section covers the key concepts and the challenges of machine learning and hybrid mechanistic/machine learning modeling.

2.3.1 Machine Learning for Bioreactor Problems

Machine learning (ML) explores the capability of computational algorithms to learn from previous large experimental data. In this context, ML employs a variety of algorithms to automate the process of data-driven models' construction, which iteratively learn to predict and improve different process outcomes (Jordan and Mitchell 2015). Several ML algorithms have been developed and are currently available in open-source python packages like *scikitlearn* (Pedregosa et al. 2011). Here we focus on ML methods that are often used in bioreactor applications. The ML area can be divided into three main classes: supervised, unsupervised, and reinforcement learning (Breve and Pedronette 2016, Nian et al. 2020, Singh et al. 2016).

Supervised learning: The supervised learning methods, such as regression (for continuous/numeric outcomes) and classification (for categoric outcomes) problems, are techniques where the task is to create a relation between a set of input/feature observations (u) and the corresponding real-valued outcome in a training dataset (y). Mathematically this relationship is described by Equation 20:

Equation 20. General relation between outputs and inputs in a ML model

$$y = f(u|\theta)$$

where *f(.)* is the model and θ the parameters. The main goal is to optimize θ to minimize the error between the model and the real values given in the training dataset (Alpaydin 2020). Figure 2 depicts a typical workflow applied in supervised learning:

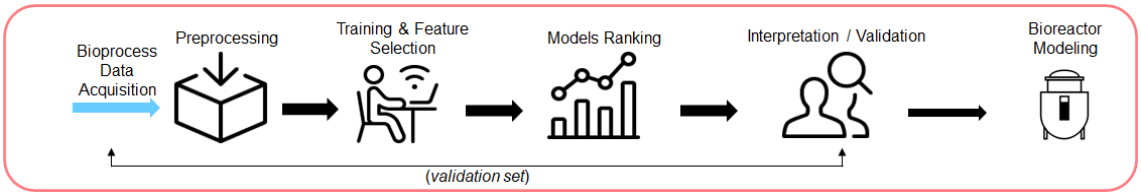

Figure 2. Typical supervised learning workflow

Unsupervised learning: Unlike supervised learning, where the data is labeled with the desired outcome value (i.e. the output associated to an observation is known), in unsupervised algorithms (learning without supervision) no pre-existing labels are required (Larranaga et al., 2006). The goal of unsupervised machine learning is to detect patterns in highly complex/multivariate input data (e.g. regions of images or search results). In other words, the basic idea is to group together similar instances using for instance the Euclidean distance. Examples of unsupervised learning techniques are K-means clustering (Yu et al. 2020) and dimensionality reduction (DR) (Butcher and Smith 2020).

Reinforcement Learning: A different paradigm regarding learning from experience is provided by reinforcement learning (RL) (Sutton and Barto 1998). RL is based on the relationship between an *agent* and the *environment*; the *agent* observes the state of the *environment* in order to take actions. For instance, in bioreactors the state observed by the agent could be the strains of microorganisms and the RL action taken by the agent control can be the concentration of substrate. The goal of an RL algorithm is to use evaluative feedback from the environment to estimate real values to update an internal policy that optimizes a desired target (Lee et al. 2018). RL can be viewed as a joint optimization problem between the policy/action and the data. Despite some challenges (e.g. satisfying operational constraints), RL can be very useful to address a wide range of chemical process control and bioprocess problems. Some examples include the nonlinear optimal control problems (Hoskins and Himmelblau 1992, Shin et al., 2019) and real-time optimization of bioreactors (Powell et al., 2020). More recently, RL has also been applied for the control of bioreactor systems (Ma et al., 2020).

In the following sections, three of the commonly used supervised learning methods in bioreactor modeling are further detailed.

2.3.1.1 Artificial Neural Networks (ANNs)

Neural network computing is currently the most popular ML tool for supervised learning in different domains including bioreactor modeling and control. ANNs are computing systems inspired by biological neural networks, consisting of multiple interconnected processing units

(called nodes) arranged in layers (Haykin, 2009). Each node is excited by the connecting nodes (typically from preceding layers) and computes an output signal that will excite other nodes in the network (typically in proceeding layers). The nodes of the first layer (input layer) receive external signals, which are propagated to the nodes of the intermediate layers (hidden layers) eventually exciting the output nodes (output layer) thereby forming the system outputs. Different ANN architectures with particular node activation functions (linear, sigmoidal, hyperbolic tangent, *ReLU*, etc...) and topologies have been proposed for different applications (Krogh 2008).

Neural network applications for bioreactor modeling and control first caught attention in the early 90s (e.g. Dimassimo et al. 1992, Dochain, et al., 1992, Joseph and Hanratty, 1993, Baughman and Liu, 1994). This surge was motivated by the publication of the error back-propagation algorithm for efficient neural network training (Rumelhart et al. 1986), which boosted neural network applications in different domains. After a long period of skepticism, they are now resurging by the new developments in deep neural network topologies and deep learning algorithms (Larochelle et al. 2009), particularly the ADAM algorithm (Kingma and Ba, 2017).

Given the non-linear character of bioreactor dynamics, the network topology most used is by far the Multilayer Perceptron Network (MLP). Particularly, a simple 3-layered MLP has been the topology of choice for now-linear regression problems in the bioreactor modeling domain. A 3-layered MLP for nonlinear regression problems consists of a linear input layer, a single hidden layer with tangent hyperbolic (tanh) nodes and an output linear layer. Mathematically, a MLP is simply stated by the following function (Equation 21):

Equation 21. Most common MLP structure for bioreactors

$$y = w_2\, tanh(w_1 u + b_1) + b_2$$

where y is the vector of outputs calculated by the output layer, $u$is the vector of inputs that excites the input layer and $\theta = \{w_1, b_1, w_2, b_2\}$ is the network parameters (weights associated with node connections) that need to be estimated from data during the training process. The MLP expressed by Equation 21 is of static nature. The extension to time series data is straightforward by considering time lagged inputs/outputs to the network. Kingma and Ba (2017) have shown how a deep MLP may be efficiently trained using the ADAM algorithm with (nodes) dropout. A network is considered deep when it has more than 2 hidden layers. This new development will likely boost new applications for bioreactor modeling and control in the near future (Salah and Fourati 2019).

2.3.1.2 Decision Trees (DTs)

The DTs (DeLisle and Dixon 2004) are a simple case of a non-parametric supervised model and can represent any function of the input attributes, with the rules defined as a tree (consists of nodes and branches). The use of a training dataset defines the set of rules that will be sequentially employed to a new observation until a class is estimated. This process goes on until a leaf (terminal) node is satisfied, corresponding to the decision outcomes (i.e., continues or categorical value) or one stopping rules are reached. The rules for each node are given by the division of the dataset producing better discriminative ability. DTs are one of the most popular algorithms due to their high human interpretability and its simplicity to implement/use. In the context of bioreactor modeling, DTs can be applied, for example, to identify critical process parameters using information from different fermentation runs (Buck et al. 2002) and to find the combination of operating variables for algal biomass and lipid production (Cosgun et al. 2021). They have also been employed to optimize fermentation medium (Bapat and Wangikar 2004).

The relationship between the outcome (y) and features (x) is described by (Equation 22):

Equation 22. Decision trees mathematical structure

$$y = \sum_{m=1}^{M} c_m \, I\{x \epsilon Rm\}$$

Here, each instance falls into exactly one leaf node (subset of Rm), c_m is the weight given to the mth transformation and $I_{\{x \in Rm\}}$ is the function that returns 1 if x is in the subset of Rm and 0 otherwise.

2.3.1.3 Random Forest (RF)

An RF is an ensemble method that combines different DTs, each with the same nodes (Breiman 2001). The RF algorithm has two main steps:(i) RF creation and (ii) make a prediction from the classifier created in step (i). This algorithm uses the sample bootstrapping aggregation method for each DTs (Rindskopf 1997). Additionally, a feature sampling is performed, making classifiers more robust to missing values and more uncorrelated to each other. For large numbers of trees, more accurate results are expected. This model prevents data overfitting and is simple to train. For a training set $T = \{(x1, y1). \ldots, (xn, yn)\}$ of N observations from random vectors (x, y), the developed RF will be an ensemble of k trees $\{t1(x). \ldots, tk(x)\}$. The ensemble produces k outputs $\{y1 = t1(x), \ldots, yk = tk(x)\}$, where y_k, $k = \{1,2, \ldots., k\}$ is the prediction for a classifier by the kth tree.

RF methods can be used as regression, classification and to assess feature importance, making it an algorithm with different applications on real-world problems. An example is given by Melcher, et al. (2015) who use RF technique to predict cell dry mass and recombinant protein based on online process parameters and spectroscopic information. Recently, RFs have been employed for on-line fault diagnosis in a bioreactor operation (Shrivastava et al. 2017).

2.4 Hybrid Mechanistic/ML Bioreactor Models

A very promising approach for bioreactor modeling is the combination of mechanistic models with ML into hybrid model structures. The combination of mechanistic models with ANNs for bioreactor dynamic modeling was first suggested by Psichogios and Ungar (1992) and Thompson and Kramer (1994). Thompson and Kramer (1994) classified this problem as a hybrid semiparametric modeling problem. Hybrid semiparametric models integrate parametric functions with fixed structure stemming from *prior* process knowledge (for example, macroscopic material balance equations) with nonparametric functions with loose structure that need to be identified from process data (for example a MLP as in the paper by Psichogios and Ungar (1992) or a radial basis function network, as in the paper by Thompson and Kramer (1994)). The main motivation was to cope with ANN disadvantages such as the inability to comply with process constraints, the tendency for data overfitting, and the poor predictive power outside the training-validation domain. The advantages of hybrid modeling may be stated in *lato sensu* as a more efficient usage of knowledge for process modeling, which ultimately translates into more accurate, precise and robust process models (Schubert et al., 1994). Many other hybrid bioreactor modeling papers followed (e.g. Preusting et al., 1996, Andserson et al., 2000, Chen et al., 2000, Galvanauskas et al., 2004, Oliveira, 2004, Teixeira et al., 2007, review by von Stosch et al. (2014). Here we focus on the general bioreactor hybrid model proposed by Oliveira (2004). This hybrid structure is formed by the general state-space macroscopic material balance Equation 3 and Equation 4.

The reaction rates term, r, is defined as a flexible mixture of parametric and nonparametric functions with the following general form (Equation 23):

Equation 23. Reaction rates term as defined by a hybrid model.

$$r = KH(c)\,\rho(c,\theta)$$

with K the yields matrix, $H(c)$a set of known kinetic rate functions (with fixed structure and known parameters; for example, Monod-type kinetics), and $\rho(c,\theta)$, a loose function with un-

known structure that needs to be identified from data. The MLP network has been the preferred ML method in the context of the general bioreactor hybrid model.

Equation 24. General form of an MLP network

$$\rho(c, \theta = \{w_1, b_1, w_2, b_2\}) = w_2 \, tanh(w_1 c + b_1) + b_2$$

The main motivation for the general hybrid model structure is to provide a flexible framework to include all reliable mechanistic knowledge in the models and to decrease the dimensionality of the ML identification problem. It explicitly assumes that macroscopic material balance equations are known *a priori* in most of the bioreactor modeling problems. The less understood part of the model in a mechanistic sense are the reaction kinetics. Thus, the experimental design and ML modeling should focus on the unknown parts, which are (some of) the reaction kinetics. In this way ML does not replace mechanistic models, it rather complements or improves existing mechanistic models.

2.5 Summary

Mathematical models are recognized as fundamental tools in chemical and biological engineering enabling to better understand process mechanisms, to reduce the experimental workload for process development, to increase the process operation robustness, to improve productivity and yield, among many other potential benefits. While process systems engineering tools have proven determinant for the development and operation of chemical processes, the penetration in the bio-industries is lagging behind. There is still today the perception that bioreactor models are more difficult to develop (higher costs) and less performing (lower benefits) in comparison to chemical reactor models. This apparently less attractive benefit/cost ratio has hampered the deployment of a consistent systems bioengineering toolbox in the bio-industries.

With the emergence of systems biology in the early 00s, several industrial cell lines have been sequenced and deeply investigated in their molecular biology traits and mechanisms. In particular, genome-scale models (GEMs) have been developed for the most important cell lines/microorganisms used in industry. While the development of GEMs for individual cell lines is work in progress, providing only a scaffold of the underlying biology, they offer the opportunity for holistic process modeling, linking cell line development, culture medium design, reactor optimization with downstream unit operations. GEMs may guide the integration of the different scales thereby realizing the concept of holistic models for process platforms.

Opposed to mechanistic modeling, data-driven modeling and Machine Learning (ML) are primarily focused on the predictive power with limited gain on process understating. The main bottleneck is the availability of thorough datasets covering the full domain of process operation. While ML in particular has enjoyed a tremendous development in the fields of image analysis and speech recognition, bioprocess applications are hindered by the scarcity of data in routine operation. The widespread use of high-throughput cultivation techniques linked with multi-data technologies will undeniably create novel opportunities for ML applications to bioprocesses. Such technologies generate large amounts of omics data, typically high-dimensional and sparse, which are difficult to integrate in bioreactor models. State-of-the-art ML algorithms offer the tools to deal with some of the faced challenges, by unraveling relationships and predictions from complex datasets without the need for *a priori* mechanistic knowledge. However, to accelerate more successful applications of data-driven approaches, high-quality bioprocess data repositories preferably in machine-readable format and new computational algorithms/tools to combine the benefits of ML and mechanistic information should be produced.

The apparently conflicting objectives between process understanding and predictive power may be mitigated by the adoption of hybrid modeling formalisms. Hybrid mechanistic/ML modeling has emerged in recent years as a promising technique for bioreactor modeling particularly in the biopharma sector. Many published studies have proven the superiority of hybrid mechanistic/ML model structures when benchmarked against the standalone mechanistic or ML model components. There are however many challenges ahead in the hybrid modeling field. Hybrid modeling has been limited to relatively simple model structures and is difficult to scale to large problems, particularly to genome scale models. With current methods it is particularly difficult to develop hybrid models with detailed mechanistic modeling of intracellular phenomena. The combination of symbolic and numeric computation frameworks will likely enable to scale-up hybrid models to more complex bioreactor problems with acceptable computation cost. The "hybridization" of GEMs and machine learning is particularly promising. Hybrid GEMs may guide the integration of the different stages of upstream and downstream processing thereby realizing the concept of holistic models for process platforms. The embedded machine learning components will confer the learning through experience feature in the realm of Industry 4.0. From our point of view, solving these formidable challenges is just possible through inter- and multi-disciplinary collaborations between academia and industry.

3

A General Deep Hybrid Model for Bioreactor Systems: Combining First Principles with Deep Neural Networks

This chapter is based on the publication: Pinto, J., Mestre, M., Ramos, J., Costa, R. S., Striedner, G., & Oliveira, R. (2022). A general deep hybrid model for bioreactor systems: Combining first principles with deep neural networks. *Computers & Chemical Engineering*, *165*, 107952.

3.1 Introduction

The first steps towards the integration of mechanistic abstraction and neural networks in process systems engineering were taken in the early 90's with the pioneering works of (Psichogios and Ungar, 1992; Su and Mcavoy, 1993; Schubert et al., 1994) and (Thompson and Kramer, 1994). The main motivation was to overcome neural networks limitations, namely the i) inability to comply with process constraints, ii) the tendency for data overfitting, and iii) the poor predictive power outside the training-validation domain. Thompson and Kramer (1994) framed this problem as hybrid semi-parametric systems, whereby parametric functions with fixed structure stemming from prior process knowledge (e.g., macroscopic material balance equations) are combined in series or in parallel with nonparametric functions (e.g. neural networks) identified from process data. Numerous bioprocess modeling studies followed (e.g. Preusting et al., 1996; van Can et al., 1998; Chen et al., 2000; Galvanauskas et al., 2004; Oliveira, 2004; Teixeira et al., 2007; Fiedler and Schuppert, 2008; von Stosch et al., 2011; Ferreira et al., 2014; Pinto et al., 2019; O'Brien et al., 2021; Bayer et al., 2021) highlighting the advantages of the hybrid technique, which may be summarized as a more rational usage of

prior knowledge eventually translating into more accurate, transparent and robust process models.

The vast majority of hybrid modeling studies explored the combination of conservation laws and shallow neural networks (see review by (von Stosch et al., 2014)). Recent advances in deep learning have however demonstrated that neural networks with multiple hidden layers (deep networks) are advantageous over their shallow counterparts. Shallow and deep networks are both universal function approximators, but deep networks are able to approximate compositional functions with exponentially lower number of parameters and sample complexity (Delalleau and Bengio, 2011; Eldan and Shamir, 2016; Liang and Srikant, 2017) and are less prone to overfitting (Mhaskar and Poggio, 2016). The shift from shallow to deep network architectures has been triggered by the development of stochastic gradient descent training algorithms, particularly the ADAM method (Kingma, 2014). ADAM is a first-order gradient-based method for stochastic objective functions based on adaptive estimates of lower-order moments. The data subsampling along with the learning rate adaptation at each iteration resulted in a simple and robust training method that is less sensitive to gradient attenuation and to the convergence to local optima. Stochastic regularization based on weights dropout has been shown to effectively avoid overfitting in deep learning (Hinton et al., 2012; Srivastava et al., 2014). Stochastic regularization is frequently associated with stochastic gradient descent methods to prevent overfitting and to improve generalization properties (Koutsoukas et al., 2017).

Only very recently the deep learning advances are penetrating the hybrid modeling field. Bangi and Kwon (2020) proposed a hybrid model for a hydraulic fracturing process that combines a First Principles model with a deep neural network. A fully connected network with 5 layers (1x20x20x20x1), hyperbolic tangent activation (*tanh*) in the 3 hidden layers and linear activation in the input/output layers, was adopted. The Levenberg–Marquardt algorithm and finite difference-based sensitivity analysis were adopted to train the hybrid model. The resulting hybrid model had superior extrapolation properties compared to a purely data-driven deep neural network model. Following a similar approach, Shah et al. (2022) developed a deep hybrid model for an industrial fermentation process. Lee et al. (2020) developed a hybrid deep model of an intracellular signaling pathway using a neural network with 2 hidden layers. Bangi et al. (2022) proposed the Universal Differential Equations (UDE) formalism for mixing the information of physical laws and scientific models with data-driven machine learning approaches. They applied it to a *Saccharomyces cerevisiae* batch fermentation process.

Merkelbach (2022) have develop a software package called HybridML that uses TensorFlow for artificial neural network training and Casadi to integrate ordinary differential equations.
In this chapter, we revisit the general bioreactor hybrid model (Oliveira, 2004; Teixeira et al., 2007; von Stosch et al., 2011; Ferreira et al., 2014; Pinto et al., 2019) and extend it to deep learning. More specifically, we explore deep learning techniques in a hybrid semiparametric modeling context, such as deep feedforward neural networks with varying depths, the rectified linear unit (*ReLU*) activation function, dropout regularization of network weights, and stochastic training with the ADAM method. These techniques are applied to two case studies and are benchmarked against the traditional shallow hybrid modeling approach.

3.2 Materials and Methods

3.2.1 General Deep Hybrid Model for Bioreactor Systems

A stirred tank bioreactor can be generically represented by the hybrid model structure of Figure 3:

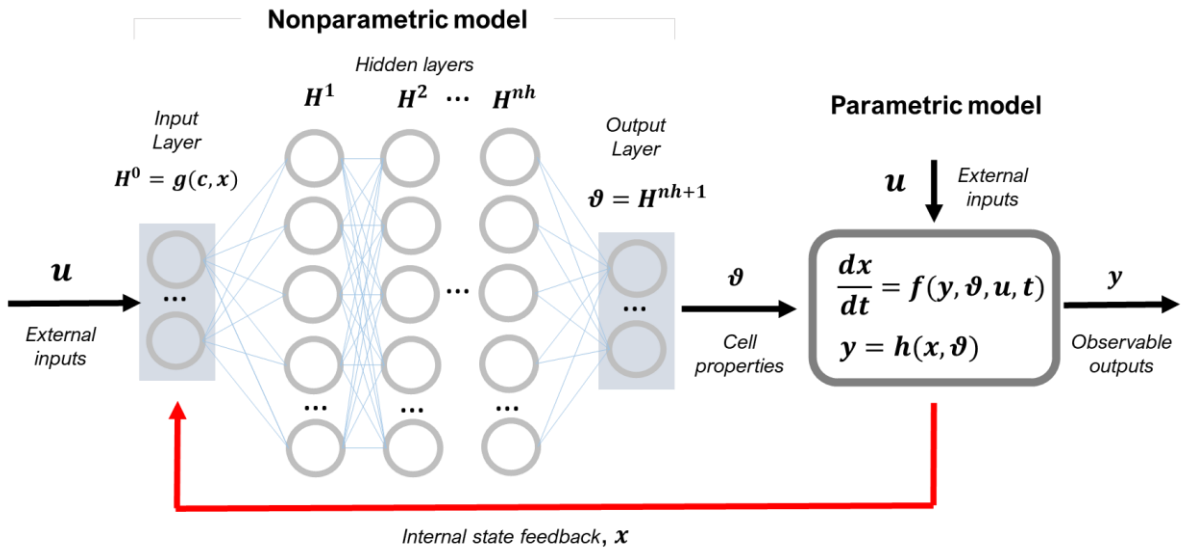

Figure 3. Schematic representation of the general deep hybrid model for bioreactor systems. The model is dynamic in nature with state vector, x, and observable outputs y. The model has a parametric component (functions f(.) and h(.)) with fixed mathematical structure determined by First Principles (typically material/energy balance equations). Some cellular properties are modelled by a deep feedforward neural network with multiple hidden layers as function of the process state, x, and external inputs, u. The deep neural network is a nonparametric model component with loose structure that must be identified from process data given the absence of explanatory mechanisms for that particular part of the model.

The dynamics of state variables are modelled by a system of ordinary differential equations (ODEs) derived from macroscopic material balances and/or intracellular material balances and/or other physical assumptions. These equations take the following general form (Equation 25 and Equation 26):

Equation 25. General ODE system of a hybrid model for a bioreactor system

$$\frac{dx}{dt} = f(y, \vartheta, u, t)$$

Equation 26. General form of the equation of observable outputs

$$y = h(x, \vartheta)$$

with t the independent variable time, $x(t)$ the process state vector, $u(t)$ the vector of external inputs (feed rates, temperature, pH, etc), ϑ a vector of process variables with unknown defining functions, and y the vector of measured variables. Equation 25 and Equation 26 are the state-space model and measurement model respectively. The functions $f(.)$ and $h(.)$ are of parametric nature thus with fixed structure stemming from prior knowledge. They are typically set by material and/or energy balance equations of extracellular and intracellular variables (as shown in the case studies). Some relevant bioprocess variables may be less defined in terms of explanatory mechanisms and/or rely on loose assumptions. Typical examples are

biological reaction kinetics or product quality attributes, which are difficult to establish on a mechanistic basis. In the general hybrid model, such properties are defined as loose functions, $\vartheta(.)$ (typically of the process state and external inputs), with unknown structure, i.e. nonparametric functions without physical meaning. Among the many possibilities to define $\vartheta(.)$, the preferred approach (in a hybrid modeling context) has been by far the feedforward perceptron networks with 3 layers only (see review by (von Stosch et al., 2014)). In the present study, the more general case of deep multi-layer perceptron networks with arbitrary number of nh hidden layers is explored, stated as follows (Equation 27, Equation 28 and Equation 29):

Equation 27. Multi-layer perceptron network input layer

$$H^0 = g(x, u, t)$$

Equation 28. Multi-layer perceptron network hidden layer

$$H^i = \sigma\left(w^i \cdot H^{i-1} + b^i\right), i = 1, \dots, nh$$

Equation 29. Multi-layer perceptron network output layer

$$\vartheta(\cdot) = w^{nh+1} \cdot H^{nh} + b^{nh+1}$$

The input layer (Equation 27) typically receives information of the state variables, x and/or external inputs, u (temperature, pH, etc.…) and/or process time, t. An optional non-linear preprocessing function $g(x, u, t)$, may sometimes facilitate the identification of $\vartheta(.)$, as for example concentration ratios are set as inputs to the neural network or other normalization rules (see (von Stosch et al., 2016; Gnoth et al., 2008; Gnoth et al., 2010)). Then follows nh hidden layers (Equation 28) with $\sigma(.)$ the nodes transfer function (in this chapter either the *tanh* or the *ReLU*). Finally, the output layer has linear nodes (Equation 29). The parameters $w = \{w^1, w^2, \dots, w^{nh+1}\}$ and $b = \{b^1, b^2, \dots, b^{nh+1}\}$ are the nodes connection weights that need to be identified from data during the training process. Presuming that initial conditions $x(t) = x_0$ and network weights $\omega = \{w, b\}$ are given, the deep hybrid model can be solved by numerical integration as an Initial Value Problem (IVP). In the present chapter, a Runge-Kutta 4th order ODE solver was adopted to integrate the system (Equation 25, Equation 26 Equation 27, Equation 28 and Equation 29) and compute x, y and ϑ over time. All the code was implemented in the HYBrid MODdeling (HYBMOD) MATLAB® toolbox on a computer with Intel(R) Core (TM) i5-8265U CPU @ 1.60 GHz 1.80 GHz, and 24 GB of RAM.

3.2.2 Training Method

3.2.2.1 Standard Non-Deep Method

Hybrid bioreactor models are typically trained by indirect supervised learning with cross-validation to avoid overfitting (e.g., (Psichogios and Ungar, 1992; Oliveira, 2004; Pinto et al., 2019; von Stosch et al., 2014)). The data are partitioned into a training subset (for parameter estimation), a validation subset (stop criterion to avoid overfitting) and a test subset (to assess the predictive power). Partitioning is typically performed batch wise with the amount of data allocated in each partition depending on the objective of the study and on the amount of data available. The optimization of network parameters is performed over the training set only in a weighted least-squares sense (Equation 30):

Equation 30. Weighted least squares for shallow hybrid modelling.

$$WSSE = \frac{1}{T}\sum_{t=1}^{T}\frac{(y_t^* - y_t)^2}{\sigma_t^2}$$

with T the number of training patterns, y_t^* the measured variables at time t, y_t^* the corresponding model prediction and σ_t the measurement standard deviation. This method is called indirect because the loss function is not directly linked to the neural network outputs, ϑ. The Levenberg-Marquardt method (LMM) has been shown to solve very effectively the indirect training problem (Equation 25, Equation 26, Equation 27, Equation 28, Equation 29 and Equation 30) in the case of shallow hybrid models (Schubert et al., 1994; Oliveira, 2004). The LMM has also been used in a recent deep hybrid modeling study (Bangi and Kwon, 2020). The LMM convergence is improved if the sensitivity equations are applied to calculate the loss function gradients instead of numerical gradients (e.g. (Psichogios and Ungar, 1992; Schubert et al., 1994; Oliveira, 2004)). The sensitivity equations for the general hybrid have the following structure (for simplicity it is assumed that ($y = x$) (Equation 31):

Equation 31. Indirect sensitivity equations for the general hybrid model

$$g = \frac{\partial WSSE}{\partial \omega} = -2\sum_{t=1}^{T}\frac{y_t^* - y_t}{\sigma_t^2}\left(\frac{\partial x_t}{\partial w}\right)$$

$$\frac{d\left(\frac{\partial x}{\partial w}\right)}{dt} = \left(\frac{\partial f}{\partial x}\right)\left(\frac{\partial x}{\partial w}\right) + \left(\frac{\partial f}{\partial x}\right)$$

$$\left(\frac{\partial c}{\partial w}\right)|_{t=0} = 0$$

The sensitivity equations are obtained by differentiation of the state-space model (Equation 25) in relation to the network parameters, w. For more details regarding the sensitivity equations in a hybrid modeling context see (Psichogios and Ungar, 1992; Oliveira, 2004). The integration of the sensitivity equations was performed in this study with a Runge-Kutta 4th order ODEs solver.

3.2.2.2 Stochastic Adaptive Moment Estimation (ADAM) With Semi-Direct Sensitivities

An important goal in this chapter is to compare the standard training method with state-of-the-art deep learning techniques in the context of hybrid modeling. Particularly, ADAM is considered a landmark in deep learning and was implemented here to train hybrid models. The ADAM method estimates the network parameters, $\omega = \{w, b\}$, through the first and second moments of the gradients of the loss function and a set of hyperparameters α, β_1 and β_2, representing the step size and exponential decays of the moment estimations (for details see (Kingma, 2014)). The loss function is the same as in the previous method (Equation 30). This results in the following implementation (Equation 32):

Equation 32. ADAM algorithm equations

$$m_k = \frac{\beta_1 \cdot m_{k-1} + (1-\beta_1) \cdot g_k}{(1-\beta_1^k)}$$

$$v_k = \frac{\beta_2 \cdot v_{k-1} + (1-\beta_2) \cdot g_k^2}{(1-\beta_2^k)}$$

$$w_k = w_{k-1} - \frac{\alpha \cdot m_k}{(\sqrt{v_k} + \varepsilon)}$$

with k the iteration number, m_k the first order moment of gradients, g_k the loss function gradients, v_k the second order moment of gradients. For the present chapter, the suggested default parameters of $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\varepsilon = 10^{-8}$ were adopted (Kingma, 2014).

The gradients at each iteration are obtained by solving the sensitivity equations (Equation 31). Because the CPU scales exponentially with the size of the network, a different approach to calculate the gradients was explored. Instead of computing the sensitivities of state variables in relation to network parameters, $\left(\frac{\partial x}{\partial w}\right)$, a semidirect approach was implemented where the sensitivities of state variables in relation to network outputs, $\left(\frac{\partial c}{\partial \vartheta}\right)$, are computed. The semidirect sensitivity equations are as follows (again assuming $y = x$) (Equation 33):

Equation 33. Semidirect sensitivity equations for the general hybrid model

$$\frac{\partial WSSE}{\partial \vartheta} = -2 \sum_{t=1}^{T} \frac{y_t^* - y_t}{\sigma_i^2} \left(\frac{\partial x}{\partial \vartheta}\right)$$

$$\frac{d\left(\frac{\partial x}{\partial \vartheta}\right)}{dt} = \left(\frac{\partial f}{\partial x}\right)\left(\frac{\partial x}{\partial \vartheta}\right) + \left(\frac{\partial f}{\partial \vartheta}\right)$$

$$\left(\frac{\partial x}{\partial \vartheta}\right)|_{t=0} = 0$$

Finally, the loss function gradients $g = \frac{\partial WSSE}{\partial \omega}$ can be computed from the $\frac{\partial WSSE}{\partial \vartheta}$ sensitivity by the well-known error backpropagation algorithm through the network (Werbos, 1974). The main advantage of the semidirect method is that the number of ODEs for calculating the sensitivities is massively reduced and are independent of the size of the network. This results in a sizable CPU reduction as shown in section 3.2.4.

3.2.3 Case Studies

3.2.3.1 *Lee & Ramirez* Synthetic Dataset

A synthetic dataset was generated based on the *Lee & Ramirez* bioreactor model (Lee and Ramirez, 1994). This model is frequently adopted as a benchmark to test different optimal control methods (e.g. (Banga et al., 2005)). The objective in this case study is to train hybrid models on an information rich dataset (time series data generated by statistical design of experiments) and then to assess if the trained hybrid models are able to describe (extrapolate) the maximum productivity fed-batch obtained by optimal control studies (Lee and Ramirez, 1994).

The *Lee & Ramirez* model describes the dynamics of biomass (X), substrate concentration (S), inducer concentration (IND), product concentration (P), shock factor (Sh), recovery factor (Re) and reactor volume (V) in a recombinant Escherichia coli fed-batch process. Experiments were simulated dynamically for different conditions (see below) applying a Runge-Kutta 4th order ODEs solver. Samples were simulated with 1 hour sampling time. Gaussian noise was added to "sampled" variables with standard deviations of 1.5 (X), 5(S) and 0.3(P) (10% of maximum concentration). As shown in section 3.2.4., modeling errors were calculated based on the noisy data (noisy weighted mean squared error (WMSE)) and also on the noise free data (noise free WSSE).

A central composite design (CCD) was applied to the process degrees of freedom, namely the induction time between 5-9 hours, pre-induction substrate feed rate between 0-0.8h^{-1}, post-induction substrate feed rate between 0-0.8h^{-1} and inducer feed rate between 0-1h^{-1}. This resulted in 25 fed-batch experiments. The 25 fed-batch experiments were included in the training data partition (297 training data points). The validation dataset (used only as training stop criterium) was obtained by adding gaussian noise with standard deviations of 1.5 (X), 5(S) and 0.3(P) to the training dataset resulting in 297 validation data points. In our experience, this partition method maximizes data usage for training and also effectively prevents model overfitting. For the test dataset (used to assess the model generalization capacity), the optimal fed-batch with optimized feeding and maximum product concentration of 3.16 g/L (Lee and Ramirez, 1994) was adopted (15 data points). In summary, the models were trained/validated with the 25 DoE experiments and then set to predict the dynamic profiles of the optimal production fed-batch. The optimal production fed-batch delivers a final product mass, which is 34.4% higher than the best DoE fed-batch. The details of the dataset are provided as supplementary material A.

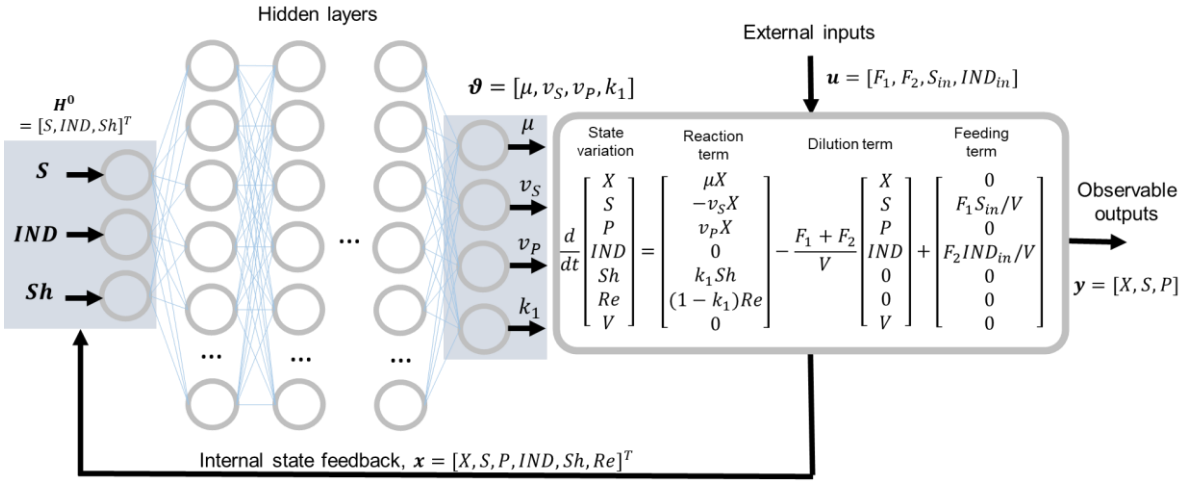

Figure 4. Deep hybrid model structure for the *Lee & Ramirez* dataset. The parametric component is established by a system of ODEs as described in Lee & Ramirez (1994). The specific biologic kinetics are considered mechanistically unknown thus modelled by a deep feedforward network. The job of this model is thus to "learn" from data the biologic kinetics under the constraint of dynamic material balance equations.

The hybrid model structure adopted for this problem is shown in Figure 4. The reactor has 7 internal sate variables $x = [X, S, P, IND, Sh, Re]^T$ of which only 3 are measured, thus $y = [X, S, P]^T$. The system of ODEs are derived from mass conservation laws and are the same as in (Lee and Ramirez, 1994). The neural network computes 4 reaction terms $\vartheta = [\mu, v_S, v_P, k_1]^T$, taken as unknown cellular features that need to be learned from data. The neural network

has only 3 inputs $H^0 = [S, IND, Sh]^T$ which were pre-selected based on prior knowledge of the reaction kinetics for this problem (Lee and Ramirez, 1994).

Hybrid models with different network depths and sizes were evaluated, with the best hybrid model discriminated on the basis of the Akaike Information Criterion with second order bias correction (AICc) computed for the training data partition as follows (Equation 34):

Equation 34. Akaike Information Criterion with second order bias correction (AICc)

$$AICc = T\ ln(WSSE) + 2\ nw + \frac{2\ nw\ (\ nw\ +\ 1\ )}{T\ -\ nw\ -\ 1}$$

AICc includes an overparameterization penalty and is commonly used to discriminate between empirical model candidates with different number of parameters, nw, and to select a parsimonious model for small sample sizes (Li et al., 2002).

3.2.3.2 MUT+ *Pichia pastoris* Pilot Dataset

A MUT+ *Pichia pastoris* expressing a single chain antibody (scFv) was cultivated in a Lab Pilot Fermenter Type LP351, 50 L, Bioengineering, Switzerland with standard instrumentation to measure on-line pH, temperature, pressure, stirrer, airflow and pO2. The wet cell weight and scFv titer were measured off-line. All the details of the experimental procedure are given elsewhere (Teixeira et al., 2006). The reactor operation is divided into three phases: glycerol batch (GB) phase, glycerol fed-batch (GFB) phase and methanol fed-batch (MFB) phase (or post-induction phase). In the GB phase, the initial glycerol level was set at 4%, taking approximately 30 h for complete depletion. Thereupon, the GFB phase starts, following an exponential feeding profile. At the end of the GFB, a transition to the MFB phase is implemented in order to minimize the adaptation time of cells to methanol. After the transition phase, the methanol feeding rate, the pH and the temperature were designed in order to generate process data to optimize scFv productivity (see (Teixeira et al., 2006) for details). A total of 9 experiments were performed with varying methanol feed rate, temperature, and pH. In this case study, only the MFB phase was considered for hybrid modeling. The dataset with the 9 experiments has 207 measurements of biomass wet cell weight in triplicate and 207 measurements of scFv in triplicate. The training-validation partition included 8 experiments and the test partition 1 experiment. All possible training-validation/test permutations were evaluated. The hybrid model structure adopted for this problem is similar to that of Figure 4 with a few adaptations (discussed in section 3.2.4.). The training and model discrimination methods were as for the *Lee & Ramirez* case study.

3.2.4 Results and Discussion

3.2.4.1 Development Of a Shallow Hybrid Model: *Lee & Ramirez* Case Study

A traditional shallow hybrid model was first developed for the *Lee & Ramirez* dataset. A shallow feedforward network with a single hidden layer with *tanh* activation function was employed. The hybrid model was trained with the standard nondeep method (LMM optimization + cross-validation + random weights initialization from the uniform distribution). The training and validation partition comprehended 25 experiments (825 training patterns) designed by statistical DoE (see section 3.2.3.1). The test partition included a single experiment with the highest protein production (optimal batch obtained by dynamic optimization as reported in (Lee and Ramirez, 1994). The test experiment has a final product mass 34.4% higher than the best training/validation experiment. For a given network size, the training was always repeated 10 times with different weights initialization between [-0.1, 0.1] and only the best result was kept (lower validation error). This procedure was repeated for hybrid models with varying number of nodes in the hidden layer keeping the same data partition and a maximum number of iterations of 20000 for comparability. The overall results are shown in Table 1:

Table 1. Training results of shallow hybrid models for the *Lee & Ramirez* data set with 25 training batches (Training WSSE), 25 validation batches (Validation WSSE) and a single test batch with the highest possible productivity obtained by optimal control (Test WSSE noisy/noise free are computed with noisy or noise free target concentrations respectively). The AICc is computed for the training data set only. Each row represents a different model with a given number of hidden nodes (between 1-15) in a single hidden layer with *tanh* activation function. The hybrid models were trained with the standard nondeep method (LMM optimization with 20000 iterations + cross-validation + random weights initialization between [-0.1, 0.1] from the uniform distribution). The training was repeated 10 times with different weights initialization and only the best result is kept for each model.

Number of hidden nodes	Training WSSE	Validation WSSE	Test WSSE (noisy)	Test WSSE (noise free)	AICc	CPU time	Number of Weights
1	20.2	20.3	42.2	2.1	2490	776	12
2	2.57	2.77	7.53	8.12	810	1320	20
3	1.16	1.31	1.08	1.39	172	1780	28
4	1.1	1.29	1.34	1.01	146	1560	36
5	2.77	3.07	6.56	5.42	922	1390	44
6	1.78	1.94	1.22	2.11	570	1730	52

7	1.40	1.70	7.87	7.31	389	1870	60
8	1.09	1.32	1.14	0.76	200	2050	68
9	1.01	1.21	1.04	0.68	150	2250	76
10	0.941	1.16	1.05	0.54	111	2250	84
11	0.949	1.22	1.33	0.83	134	2360	92
12	0.914	1.11	0.86	0.75	121	2290	100
13	0.935	1.07	1.03	0.69	154	2280	108
14	0.944	1.15	1.10	0.93	183	2230	116
15	0.899	1.11	0.937	0.62	152	2670	124

From these results, it is possible to conclude that the optimal number of hidden nodes is 10 corresponding to the lowest corrected Akaike information criterion (AICc) value (111). Of note, the AICc criterion, which is calculated for the training partition only, coincided with the lowest noise free test error (0.54 noise free WSSE; to note that the noise free WSSE is computed on process data uncorrupted by experimental noise, thus a better metric for accessing the predictive power). Despite the coincident outcome in this case, the AICc sometimes fails to discriminate the structure with the highest predictive power as shown in the next sections. Moreover, the noisy test error of the selected model with 10 hidden nodes (noisy WSSE=1.05) is only moderately higher (11,6%) than the corresponding training error (WSSE=0.941).

3.2.4.2 Comparing The Deep and Shallow Hybrid Modeling Approaches

Several hybrid structures with varying neural network depths (2-4 hidden layers) were compared with the shallow network case (1 hidden layer). The same *Lee & Ramirez* dataset and data partition were kept as in the previous section. We first focused on the *tanh* activation (in the hidden layers), which has been the standard for nonlinear regression problems with shallow neural networks (Cybenko, 1989). Every model structure was trained with two different methods: the traditional LMM+CV+*tanh* and ADAM+CV+*tanh*. The training was always repeated 10 times and only the best solution (lowest validation error) was kept, as before. The number of iterations for the ADAM method was 20000 as for the LMM method. The overall results are shown in Table 2:

Table 2. Comparison of deep and shallow hybrid models for the *Lee & Ramirez* data set (same data partition as in Table 1) trained either by the LMM algorithm or by the ADAM algorithm. In all cases cross-validation (CV) and indirect sensitivities were applied. Each row represents a different shallow or deep hybrid model structure using either *tanh* or *ReLU* in the hidden layers. The training was repeated 10 times with different weights initialization and only the best result is kept.

Hybrid model	Training method	Hidden layer type	Training WSSE (noisy)	Valida-tion WSSE (noisy)	Testing WSSE (noisy)	Testing WSSW (noise free)	AICc	CPU time	Weights
Shallow 5	LMM+CV	*tanh*	2.77	3.07	6.56	5.42	922	1390	44
Shallow 10	LMM+CV	*tanh*	0.941	1.16	1.05	0.54	111	2250	84
Deep 5x5	LMM+CV	*tanh*	1.06	1.31	1.40	1.05	198	1674	68
Deep 5x5x5	LMM+CV	*tanh*	0.921	1.17	1.13	0.72	154	74892	98
Deep 5x5x5x5	LMM+CV	*tanh*	0.835	1.09	0.915	0.32	155	81430	128
Shallow 5	ADAM+CV	*tanh*	1.22	1.32	1.20	0.66	242	33476	44
		ReLu	1.02	1.05	1.03	0.35	98	33410	
Shallow 10	ADAM+CV	*tanh*	1.60	1.21	0.91	0.24	547	30376	84
		ReLu	1.34	1.13	0.94	0.14	352	30200	
Deep 5x5	ADAM+CV	*tanh*	0.937	1.15	0.82	0.14	95	28567	68
		ReLu	0.926	1.08	0.923	0.05	90	28122	
Deep 5x5x5	ADAM+CV	*tanh*	0.936	1.16	0.81	0.09	168	32285	98
		ReLu	0.886	1.04	0.96	0.04	87	32174	
Deep 5x5x5x5	ADAM+CV	*tanh*	0.870	1.11	1.05	0.28	189	40570	128
		ReLu	0.841	1.07	0.942	0.16	152	40514	

The results in Table 2 clearly show ADAM to outperform the LMM method in what concerns the predictive power of the final model (the noise free test WSSE; to note that the AICc is not

an adequate metric to compare models of equal sizes). This conclusion is valid for deep or shallow hybrid structures. The best shallow structure with 10 hidden nodes (identified in the previous section) improved the noise free test error from 0.54 to 0.24 (>2-fold decrease) with ADAM+CV+*tanh*. The same conclusions can be taken for the deep structures, without exception. The key conclusion is that the ADAM method systematically increases the predictive power of the final hybrid model for the *Lee & Ramirez* data set.

The best model (with *tanh* activation function) among the deep and shallow structures is the 5x5x5 deep hybrid model with 98 weights, showing a noise free test error (WSSE = 0.09) 2.7-fold lower than the best hybrid shallow case (WSSE=0,24). The AICc miss spotted the best deep model. It identified the 2nd best model (5x5 structure) with, however, comparable performance. In terms of CPU, the ADAM method is generally more expensive than the LMM method for small size networks. This pattern reverses for large size networks (e.g. the best 5x5x5 structure decreased CPU by 2,3-fold with ADAM in comparison to LMM). Thus, the CPU scales more steeply with the network size in the case of LMM training when compared to ADAM training. This favors ADAM for deep hybrid structures, both in terms of predictive power and CPU time for training.

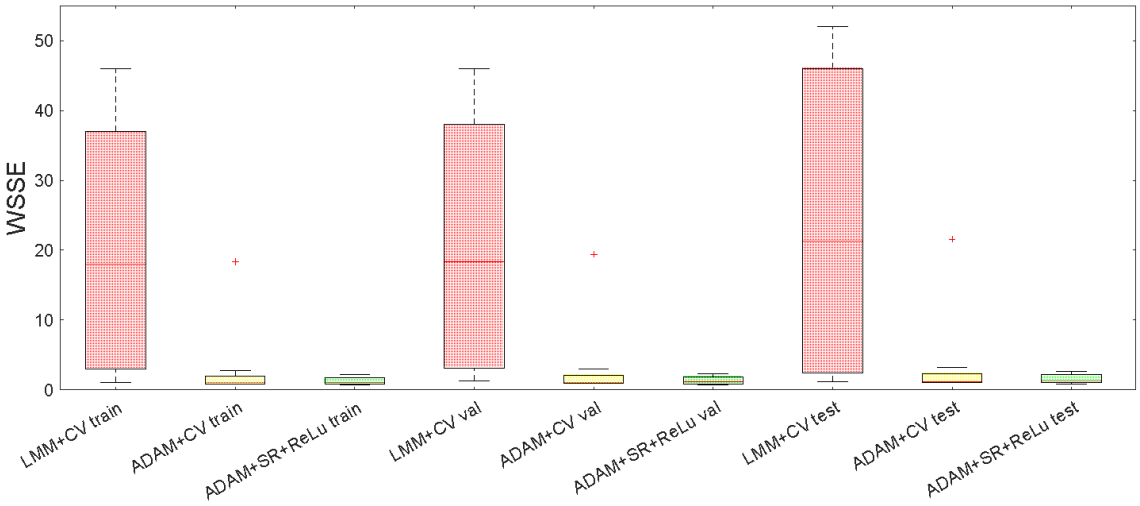

Figure 5. Boxplot of training, validation and testing WSSE for 10 training repetitions of the deep hybrid structure 5x5x5 trained by different training approaches either using the LMM or the ADAM method. Ten sets of initial weights were randomly generated (one per repletion) and kept the same in all tests performed for comparability.

Figure 5 shows the effect of weights initialization on the final training, validation, and testing error for the best deep configuration 5x5x5 when the model is trained with LMM or with ADAM. The initial weights values were kept the same for LMM and ADAM training for comparability. Interestingly, the dispersion of the errors for 10 repetitions with different weights ini-

tialization is significantly lower for ADAM in comparison to LMM, irrespective of the data partition (train, validation, or testing). There is an outlying point with significantly higher final errors for both the LMM and ADAM trainings. Concordant results were obtained for the other model configurations (results not shown). This suggests the ADAM method to be less sensitive to weights initialization. Similar conclusions were reported by (Hiscock, 2019) for standalone deep neural networks, who showed that gradient descent training methods with variable learning rate (such as the ADAM method) are less prone to be trapped in local optima thus less sensitive to weights initialization. The key conclusion to be taken is that the number of repetitions for different weights initialization may be mitigated in the case of ADAM training. This represents a potential 10-fold cut in CPU time in comparison to the LMM method for the case of 10 repetitions.

The *ReLU* activation function in the hidden layers has been a key achievement in deep learning, outperforming the *tanh* function for standalone deep neural networks (Nair and Hinton, 2010). The use of *ReLU* was investigated comparatively with *tanh* in a hybrid modeling context. Table 2 compares hybrid model performances using the one or the other activation function in the hidden layers trained by ADAM + CV using the same training procedure. The key conclusion to be taken is that the *ReLU* further improved the training and test error in all cases without exception. The best 5x5x5 structure further decreased the noise free test WSSE from 0.09 (with *tanh*) to 0.04 (with *ReLU*) at comparable CPU cost. Our results clearly show the *ReLU* to be advantageous in a deep hybrid modeling context as previously shown for (standalone) deep neural networks (Nair and Hinton, 2010). The *ReLU* activation function was thus adopted in all proceeding studies. These results might be related to the problem of gradients vanishing/exploding in deep networks. Typically, the *tanh* activation function is associated with vanishing gradients whereas the *ReLU* is associated with exploding gradients (Ding, et al., 2018) (Ding et al, 2018). The ADAM training is invariant to diagonal rescaling of the gradients. It does not completely avoid the problem of gradient vanishing when *tanh* is used. The use of ADAM with *ReLU* is however very efficient at avoiding gradient explosion since it performs dynamic scaling of the learning rate (down) when the gradients become very large.

3.2.4.3 Introducing Stochastic Regularization

Stochastic regularization (SR) has been reported as an effective method to avoid overfitting in deep learning (Srivastava et al., 2014). Here we study the ADAM method with stochastic regularization in replacement of the cross-validation technique. More specifically, ADAM was implemented with the minibatch technique and the weights dropout technique. The mini-

batch technique consists of a random selection of the training patterns from the uniform distribution using a cutoff probability parameter. Similarly, the weights dropout technique used random weights selection according to a cutoff probability parameter.

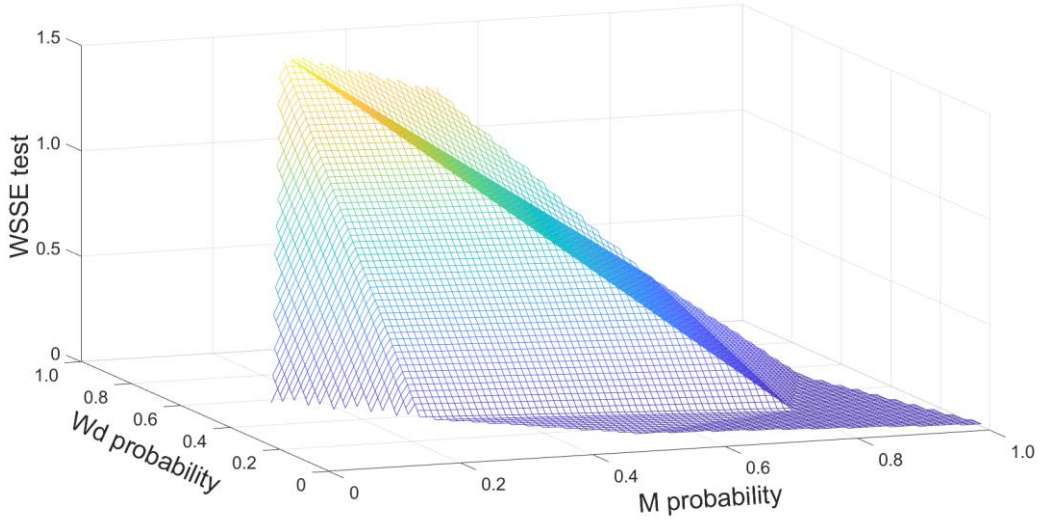

Figure 6. Effect of stochastic regularization (SR) on the predictive power of the hybrid model configuration 5x5x5 trained with ADAM + SR + indirect sensitivities with 20000 iterations for the *Lee & Ramirez* data set. Obtained noise free test WSSE over minibatch probability (M probability) and weights dropout probability (Wd probability).

Figure 6 shows the lowest WSSE test among the 10 repetitions as function of the minibatch size probability and of the weights dropout probability. The training performance is indeed very sensitive to the choice of these two parameters. The optimal minibatch probability is ~90% and the optimal dropout probability is ~50%. The final noise free test WSSE was 0.0258, which is 35.5% lower than the corresponding solution without stochastic regularization (Table 2, ADAM+CV+*ReLU*). The final train and test errors among the 10 repetitions are shown in Figure 5. Interestingly the stochastic regularization eliminated the outlying training result obtained by LMM+CV and ADAM+CV in the previous section. This result is promising because it shows the weights initialization to have practically no influence on the final training outcome. If repetitions are not needed, the CPU cost may be significantly reduced in relation to the LMM+CV or ADAM+CV methods.

3.2.4.4 Speeding up Hybrid Deep Learning by Semidirect Sensitivities

The results above support ADAM + deep networks + stochastic regularization to produce hybrid models with higher predictive power in comparison to the traditional shallow hybrid approach. Nevertheless, deep models tend to have large networks with the CPU time increasing with the network size (Luo et al., 2005). Solving the sensitivity equations is responsible for

a significant part of the CPU cost. Taking the 5x5x5 hybrid structure as example, solving the sensitivity equations implies integrating $98 \times 5 = 490$ ODEs along with the hybrid model ODEs for the computation of the objective function and objective function gradients. Such a large number of ODEs represents a significant CPU burden. A different implementation of the sensitivity method was investigated, namely the semidirect sensitivity equations (see section 3.2.2.2) in an attempt to reduce CPU time. In the semidirect approach, a much lower number of $\left(\frac{\partial c}{\partial v}\right)$ sensitivity equations are integrated over time. For the same 5x5x5 hybrid structure, the $\left(\frac{\partial c}{\partial v}\right)$ sensitivities only require $5 \times 4 = 20$ ODEs to be integrated over time. Furthermore, the semidirect sensitivity equations are independent of the number and size of hidden layers (they depend only on the number of network inputs and outputs).

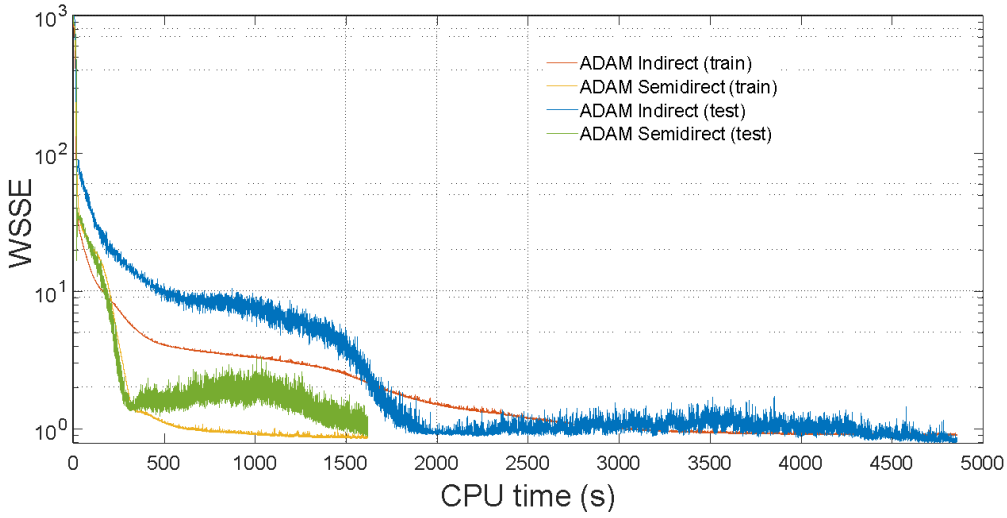

Figure 7. Training and testing error (WSSE) over CPU time for 1) shallow hybrid model {10} + LMM +CV with ten repetitions (blue line) 2) the hybrid model 5x5x5 trained with ADAM + stochastic regularization + indirect sensitivities (red line) and 3) ADAM + stochastic regularization + semidirect sensitivities (yellow line)

Figure 7 shows the variation of the train and test cost function over CPU for the configuration 5x5x5. This result shows that the semidirect sensitivity equations produced a comparable final training WSSE in relation to the indirect sensitivity equations. The convergence is however much faster. The CPU time could be reduced by 77.4% when adopting the semidirect sensitivity equations in comparison with the indirect approach. Furthermore, the test error follows similar patterns for both methods reaching a comparable final value.

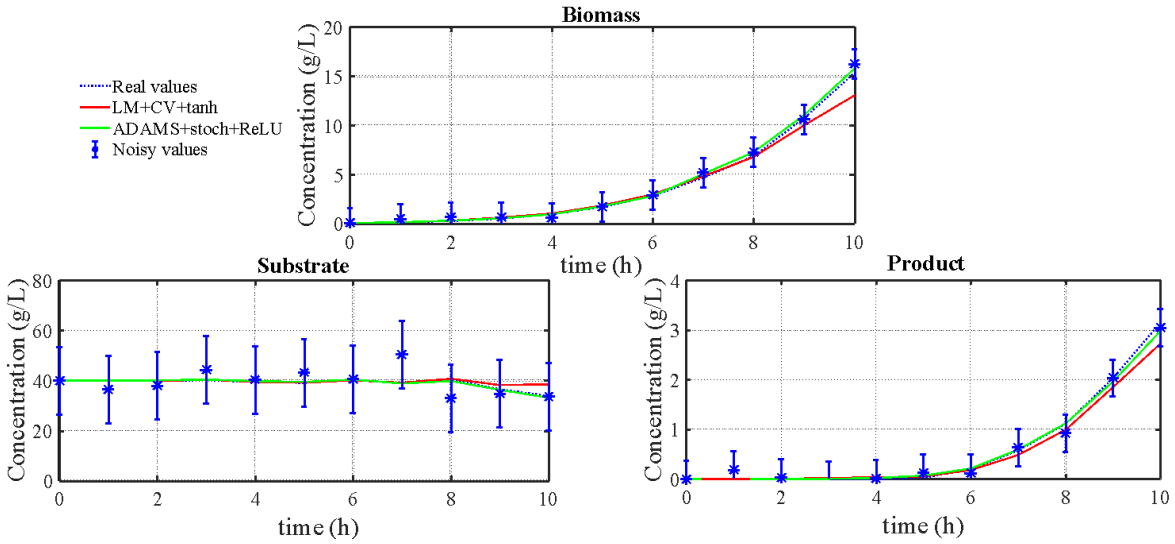

Figure 8. Prediction of the dynamic profiles of observable variables (biomass -X, substrate– S and product–P) of the test batch (*Lee & Ramirez* dataset) by the best shallow hybrid model trained with the standard method (10 hidden nodes) and by the best deep hybrid model (5x5x5). Asterisks represented observations and respective ± standard deviation. The dashed line represents the "true" noise-free process behavior (hidden to the training of the hybrid models). The red line represents the predictions of the shallow hybrid model. The green line represents the prediction by the deep hybrid model. The shallow hybrid model used the *tanh* function and was trained by the traditional non-deep method (LMM algorithm + CV + indirect sensitivities + 10 repetitions and only the best result is kept). The deep hybrid model used the *ReLU* activation function and was trained by the novel method (ADAM + SR + semidirect sensitivities + no repetitions).

Figure 8 shows the prediction of the optimal batch dynamics by the hybrid 5x5x5 model trained with ADAM+SR+*ReLU*+semidirect compared to the standard shallow model with 10 hidden nodes (LMM+CV+*tanh*+indirect). The noise free test WSSE was 0.03 and 0.54, respectively (94.4% reduction). It may be seen that both models are able to describe fairly well the dynamics of the test experiment up to 7.5 hours. There are however some visible differences towards the end of the cultivation. The shallow hybrid model underestimated the final biomass and final product by 15.3% and 13.8% respectively, whereas the deep hybrid model overestimated the final biomass by 2.7% and underestimated the final product by 5.8% only.

3.2.4.5 Pilot Scale *Pichia pastoris* Case Study

Hybrid models were developed for the *P. pastoris* process with a similar structure to the *Lee & Ramirez* model. The biomass and product material balance equations, and the shock factor ODEs are kept the same in both models. A few modifications were however required as follows:

- The inducer material balance equation was removed because in the MUT+ *P. pastoris* expression system the methanol is simultaneously the main carbon source and the inducer of foreign protein expression.
- The substrate material balance equation was also removed because methanol concentration (the substrate) was not measured. This is a limitation imposed by the experimental protocol. Instead, the measured volumetric methanol feed rate (F_{met}, g/Lh) and the measured total methanol fed to the reactor (g) were set as external inputs to the neural network.
- Temperature (T) and pH were also added as external inputs to the neural network as these two parameters varied between 17.2-30.1ºC and pH 4.0-7.0 in the experiments performed as part of a design of experiments to study the influence of these two parameters in the protein expression.
- The neural network computed the volumetric protein production rate (output) instead of the specific protein production rate as in the case of *Lee & Ramirez*. It is known that *Pichia pastoris* secretes proteases that hydrolyses the target product on certain experimental conditions (Cereghino and Cregg, 2000). The neural network is thus set to calculate the apparent volumetric production rate of the scFv, which lumps the synthesis and hydrolysis in the same kinetic term.

We have investigated the optimal hybrid structures and concluded that the two best shallow and deep hybrid structures previously identified for the *Lee & Ramirez* case study (namely the shallow structure with 10 nodes in the hidden layer and the deep 5x5x5 structure) also apply for the *Pichia pastoris* case study (results not shown). The number of parameters in both the shallow and deep models is the same, namely 123. The shallow hybrid structure was trained with the traditional method (LM+CV+*tanh*+direct, 10 repetitions with random weights initialization from the uniform distribution) whereas the deep hybrid structure was trained with the new method (ADAM+SR+*ReLU*+semidirect, weight dropout probability of 0.5, minibatch probability of 0.9 and no repetitions). Eight reactor experiments were used for training-validation (validation data points were obtained by adding gaussian noise to the training data points as in the *Lee & Ramirez* case study) and just one experiment for testing. All possible training-validation/testing permutations were evaluated. The overall results are shown in Table 3 where each row represents a different training-validation/testing permutation:

Table 3. Comparison of deep and shallow hybrid models for the pilot reactor MUT+ *Pichia pastoris* dataset. Each row represents a hybrid model obtained by training over a different training/testing data permutation (Test batch

ID refers to the batch used for testing while the remaining 8 batches were used for training/validation). Shallow hybrid models had *tanh* activation function and were trained by the traditional non-deep method (LMM algorithm + CV + indirect sensitivities + 10 repetitions and only the best result is kept). Deep hybrid models used the *ReLU* activation function and were trained by the novel method (ADAM + SR + semidirect sensitivities + no repetitions).

Test batch ID	Model type	Training WSSE (noisy)	Testing WSSE (noisy)	AICc	CPU time
F037	Shallow 10	2.18	2.58	664	19560
	Deep 5x5x5	1.79	2.13	587	13980
F044	Shallow 10	2.42	3.94	700	46440
	Deep 5x5x5	2.14	3.73	633	19980
F048	Shallow 10	2.01	2.55	626	15060
	Deep 5x5x5	1.96	2.28	618	12000
F061	Shallow 10	2.65	4.69	738	22860
	Deep 5x5x5	1.98	4.05	620	14520
F066	Shallow 10	2.54	2.82	722	13680
	Deep 5x5x5	1.59	1.86	542	9660
F007	Shallow 10	2.79	4.13	752	23640
	Deep 5x5x5	2.24	2.98	663	13320
F009	Shallow 10	2.82	4.72	754	30180
	Deep 5x5x5	2.62	3.76	730	12480
F018	Shallow 10	2.48	3.28	710	15900
	Deep 5x5x5	2.31	2.67	684	10200
F072	Shallow 10	3.15	4.85	791	26100
	Deep 5x5x5	3.01	3.98	775	14820

Sum	**Shallow 10**	**23.04**	**33.6**	**6457**	**213420**
	Deep 5x5x5	**19.64**	**27.4**	**5852**	**120960**

As an illustrative example, Figure 9 shows the measured and predicted dynamic profiles of biomass and product for the case of experiment F66 used for testing:

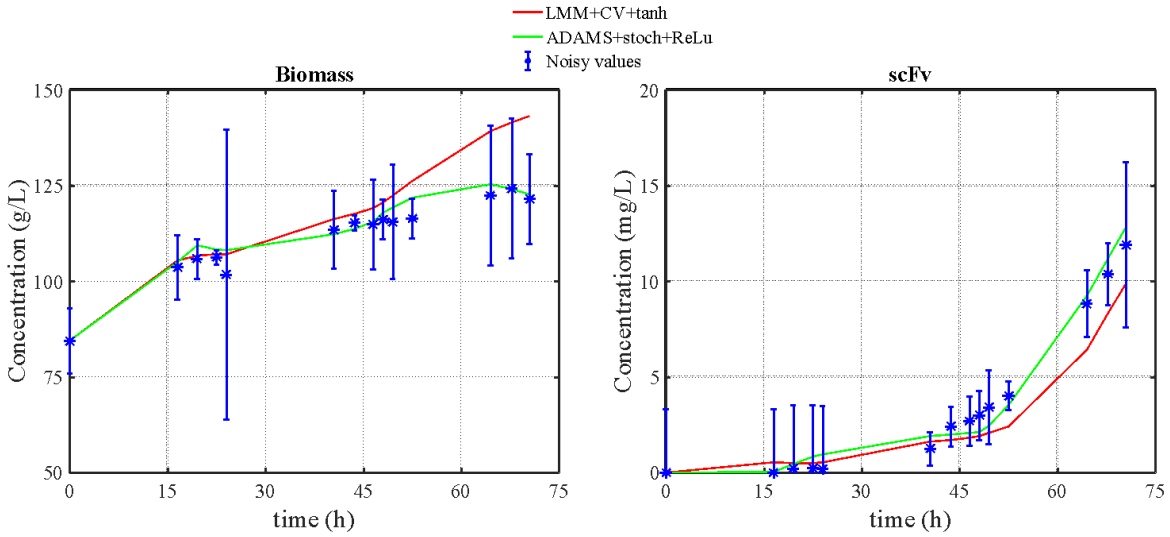

Figure 9. Prediction of the dynamic profiles of observable variables (biomass-X, and product–scFv) by the shallow (10) hybrid model and by the deep (5x5x5) hybrid model for the test batch F066 of the MUT+ *Pichia pastoris* pilot data set. Asterisks represent observations and respective ± standard deviation. The red line represents the predictions of the shallow hybrid model. The green line represents the prediction of the deep hybrid model. The shallow hybrid model used the *tanh* activation function and was trained by the traditional non-deep method (LMM algorithm + CV + indirect sensitivities + 10 repetitions and only the best result is kept). The deep hybrid model used the *ReLU* activation function and was trained by the novel method (ADAM + SR + semidirect sensitivities + no repetitions).

The key conclusions to be taken is that both the training and testing WSSEs were lower for the deep hybrid structure in relation to the shallow structure, in all data partitions tested without exception. The AICc criteria also points to the same conclusion. The differences between the dynamic profiles of biomass and scFv are clearly visible in Figure 9. The predicted final scFv titer by the shallow hybrid model is 17.5% below the experimental value whereas the deep hybrid model overestimated the experimental value by only 4.2%. Taking all data partitions together (last row in Table 3), the average training WSSE decreased by 14.8% whereas the average testing WSSE decreased by 18.4% for the deep hybrid structure in relation to the shallow hybrid structure. Moreover, the average CPU time decreased by 43.4% when applying the deep methodology in comparison to the standard methodology.

3.2.5 Conclusions

In this chapter the general bioreactor hybrid model was revisited, and some recent deep learning techniques were investigated in the context of hybrid modeling. The effect of increasing the depth of the neural network resorting to two different training approaches was investigated. The traditional approach uses the Levenberg-Marquardt optimization coupled with the indirect sensitivities, cross-validation, and *tanh* activation function. The novel hybrid deep approach uses the adaptive moment estimation method (ADAM), semidirect sensitivities, stochastic regularization and *ReLU* activation functions in the hidden layers. Two applications were addressed, one with a synthetic data set, the other with an experimental dataset collected in a pilot 50 L bioreactor. The key conclusion to be taken is that there is a clear advantage of adopting hybrid deep models both in terms of predictive power and in terms of computational cost in relation to the shallow hybrid case. In the *Lee & Ramirez* case study, the prediction error decreased 94.4% and the CPU decreased 29%. In the case of the *P. pastoris* case study, the prediction error decreased 18.4% and the CPU decreased 43,3%. The ADAM method coupled with stochastic regularization shows two significant advantages. First, it is practically insensitive to weight initialization thereby eliminating the need for training repetitions. Second, the stochastic nature of the method is less sensitive to experimental noise, eliminating the need for cross-validation. Lastly, the introduction of semidirect sensitives further decreases the CPU time particularly for large deep structures as the number of sensitivity equations (that need to be integrated over time) becomes independent of the number of hidden layers.

4

HYBRID DEEP MODELLING OF A GS115 (MUT+) *PICHIA PASTORIS* CULTURE

This chapter is based on the publication: Pinto, J., Ramos, J. R., Costa, R. S., & Oliveira, R. (2023). Hybrid Deep Modeling of a GS115 (Mut+) Pichia pastoris Culture with State–Space Reduction. *Fermentation*, *9*(7), 643.

4.1 Introduction

Many biomanufacturing companies are currently investing in digitalization tools such as big data analytics and digital twins (Udugama et al. 2021). Big data analytics applies artificial intelligence techniques on large collections of both structured and unstructured biological and process data. Such large volumes of heterogeneous data are processed by machine learning techniques such as artificial neural networks, deep learning, support vector machines, random forest, and many others, to extract valuable process insights (Yang et al. 2023). Digital Twins (DT) rely on high-fidelity mathematical models with different levels of integration with the physical process. A fully-fledged DT applies a mathematical model that receives information from the physical process in real-time and also manipulates the process in real time (Udugama et al. 2021, Appl et al. 2021). In its simplest form, a DT consists of a thoroughly validated mathematical model with historical data that is able to produce high-fidelity simulations of the physical process thus allowing to conduct in silico experiments in replacement of the physical process (Lukowski, Rauch, and Rosendahl 2019).

Many authors are considering the combination of mechanistic models with machine learning in hybrid modeling workflows for bioprocess digitalization (Badr and Sugiyama 2020). Hybrid modeling naturally pops up as a digitalization framework as it allows to integrate prior mech-

anistic knowledge with large volumes of process data in a straight-forward way. Hybrid modeling is a well-established framework in process systems engineering (von Stosch et al. 2014) and in bioprocessing (Agharafeie et al. 2023). It has covered a wide range of biological systems applications for process measurement, monitoring, optimization, and control, which are the basic building blocks of a bioprocess DT (Udugama et al. 2021).

The *P. pastoris* yeast has evolved to an industrial workhorse for microbial production of recombinant proteins (De Brabander et al. 2023). However, only a few studies have addressed hybrid modeling of *P. pastoris* cultures. Ferreira et al. (2014) developed a simple hybrid model of *P. pastoris* GS115 (Mut+) based on a shallow feedforward neural network (FFNN) combined in series with macroscopic material balance equations. The shallow FFNN described the specific growth rate and specific product synthesis rate as a function of reactor pH, temperature and volumetric methanol feeding rate. An iterative batch-to-batch control scheme was applied to optimize methanol feeding, pH and temperature based on the hybrid model resulting in a fourfold titer improvement after 4 optimization cycles. Brunner et al. (2020) developed a soft sensor based on a hybrid model that combined a carbon balance model (mechanistic) and a multilinear regression model (statistical) for the prediction of biomass concentration in real time. The software sensor was able to adapt automatically between glycerol and methanol feeding. Pinto et al. (2022) have recently applied a deep learning technique to a hybrid model of a *P. pastoris* process. FFNNs networks with 2-3 hidden layers were combined in series with material balance equations and trained with a deep learning technique, namely the adaptive moment estimation method (ADAM), semidirect sensitivity equations and stochastic regularization. The main outcome was an increase in the prediction accuracy by 18.4% and a decrease of CPU training time by 43.4% in comparison to shallow hybrid modeling.

Previous *P. pastoris* hybrid modeling studies have considered only a few state variables due to the very simple culture medium employed. Indeed, *P. pastoris* is capable of growing in a chemical defined media containing a carbon source (e.g. glycerol and/or methanol (Met)), a nitrogen source (ammonium (NH4)) and a few essential inorganic elements (Zhang and Greasham 1999). However, inorganic elements also play an important role in cell physiology. Magnesium (Mg), calcium (Ca), potassium (K), copper (Cu), strontium (Sr), iron (Fe), zinc (Zn), manganese (Mn) and chloride (Cl) were reported to be essential elements for yeast (Spencer 1997). None of the previous hybrid modeling studies have analyzed the effect of inorganic elements dynamics on recombinant protein production by *P. pastoris*. Metal ions serve as structural components of proteins and metalloenzymes and as structural elements of enzyme

active sites (Plantz et al. 2007). Magnesium (Mg), Mn and Ca are cofactors of several enzymes present in yeast, such as ATPases (Willsky 1979), (Okorokov and Lehle 1998), aspartases (Depue and Moat 1961) and glycolytic enzymes (Walker and Maynard 1997). Potassium(K) and Na are key elements in the regulation of electrochemical gradients in yeast (Arino, Ramos, and Sychrova 2010) (Martinez-Munoz and Pena 2005). The inclusion of Zn, Co, and Mn in the *P. pastoris* medium was shown to affect the quality of the final product, namely the activity of a recombinant phospholipase C (PLC) (Seo and Rhee (2004)).

In this chapter, a hybrid deep modeling framework was applied to describe the cultivation dynamics of a GS115 (Mut+) *P. pastoris* strain expressing a scFv fragment. Cultivation data acquired in a pilot 50 L bioreactor in Basal Salts Media (BSM) (Invitrogen) under different conditions of methanol feeding, temperature and pH were analyzed. The BSM medium is probably the most frequently used medium for high cell density *P. pastoris* cultivation. It contains high concentrations of phosphorus (P), sulfur (S), Ca, Mg, and K to support high cell density (Brady et al. 2001, Cereghino et al. 2002, Damasceno et al. 2004). Precipitation of BSM salts has been however reported during BSM handling at pH higher than 5.0 (Cos et al. 2006, Ghosalkar, Sahai, and Srivastava 2008). In this study, the inorganic elements were assayed during cultivation by inductively coupled plasma-atomic emission spectroscopy (ICP-AES). A hybrid deep model was developed to describe process dynamics as a function of control inputs. A key difference from previous studies (Pinto et al. 2022) is the inclusion of inorganic elements dynamics in the state-space vector. Since the mechanisms underlying the biological kinetics of inorganic elements are not well understood, the hybrid mechanistic/FFNN approach was adopted.

4.2 Materials and Methods

4.2.1 Strain, Medium and Inoculum Preparation

A genetically engineered GS115 (Mut+) *P. pastoris* strain expressing a scFv was used in this study. The Basal Salts Media (BSM) was used during cell stocking, in cryogenic vials at $-80°C$, and all cultivation steps (pre-inoculum, inoculum and bioreactor). The BSM solution was formulated and sterilized at 121°C for 30 minutes containing: H_3PO_4 85%, 26.70 ml/L, $CaSO_4.2H_2O$ 0.93 g/L, $K_2SO_4$18.20 g/L, $MgSO_4.7H_2O$ 14.90 g/L, KOH 4.13 g/L and glycerol 40.00 g/L. The Pichia Trace Metal (PTM1) solution was formulated as follows: $CuSO_4.5H_2O$ 6.00 g/L, NaI 0.08 g/L, $MnSO_4.H_2O$ 3.00 g/L, $Na_2MoO_4.2H_2O$ 0.20 g/L, H_3BO_3 0.02 g/L,

$CoCl_2.6H_2O$ 0.50 g/L, $ZnCl_2$ 20.00 g/L, $FeSO_4.7H_2O$ 65.00 g/L, H_2SO_4 5.00 ml/L and biotin 0.20 g/L. The PTM1 solution was filter sterilized, using a 0.22 mm pore size filter, and added to the temperature sterilized BSM solution at a volumetric ratio of 4.35 ml.L-1. The pH of the BSM solution was adjusted to pH=5.0 with 25% ammonium hydroxide. The pre-inoculum was composed of 40 mL BSM pH 5.0 and 1 mL of cell stock. The pre-inoculum was incubated at 30°C at 150 rpm for 3 days. The bioreactor inoculum consisted of 10 mL of pre-inoculum and 750 mL BSM pH 5.0. It was incubated for 3 days at 30°C and at 150 rpm.

4.2.2 Bioreactor Operation

A Lab Pilot Fermenter Type LP351, 50 L, with 42 L working volume (Bioengineering, Wald, Switzerland) was used in all bioreactor experiments. The initial bioreactor volume was 15L of BSM pH 5.0. The aeration rate and overhead pressure were 1800 L.$h^{-1}$and 100 mbar respectively at the beginning of operation. The cultivation started at 300 rpm stirrer speed. The reactor was inoculated with 750 mL of pre-inoculum. Then the process undergoes two distinct phases using two distinct substrates. The first phase is the glycerol batch/fed-batch (GBFB) phase. It starts in batch mode for approximately 30 h with an initial glycerol concentration of 40 g/L. Once the glycerol is nearly depleted, the glycerol fed-batch starts with an exponential feeding profile for approximately 12 hours to increase cell density. The cell density reached at the end of the GBFB phase varied depending on the glycerol feeding program. The second phase starts with methanol induction by the addition of 20 g/h to 100 g/h (depending on experiment) of methanol for 5 hours. A smooth transition between glycerol and methanol is applied to minimize the adaptation time to methanol metabolization. It then followed a methanol fed-batch (MFB) phase with a feed program that varied in the experiments. The temperature and pH were controlled to different set points depending on the experiment. It was not possible to control temperatures below 23.6°C due to heat transfer limitation. The pH was controlled with the addition of ammonium hydroxide 25%. The dissolved oxygen (pO2) starts at ~100% at the inoculation point and then decreases as biomass grows. Once it reaches 50% of saturation, a PID (Proportional-Integral-Derivative) controller is started to manipulate the stirrer between 300 to 1000 rpm and then the pressure between 100 to 800 mbar in order to regulate pO2 to a constant 50% set point. Further details on the experimental protocol are provided elsewhere (Ferreira et al. (2014)).

4.2.3 Analytical Techniques

Samples were withdrawn from the bioreactor at regular intervals for off-line analysis at a frequency of 4-6 samples per day. The optical density was measured in a spectrophotometer at 600nm (OD600) after appropriate dilution of the broth ensuring a value within the linear range (<0.6). For the determination of wet cell weight per unit volume (gWCW/L), samples of the culture broth were taken in triplicate and centrifuged at 15000 rpm for 10 min at 4°C. The centrifuged cell pellets were weighted to determine the sample wet cell weight (WCW). The secreted scFv was assayed by Enzyme-Linked Immuno-sorbent Assay (ELISA) according to the protocol described in Ferreira et al. 2012. The concentration of inorganic elements (P, K, Mg, S and Ca) in supernatant samples were assayed by inductively coupled plasma-atomic emission spectroscopy (ICP-AES). The conditions of the ICP-AES system were the following: Argon with the flow 15 L. min^{-1}, temperature between 5700–10000°C, pressure of 3 bar and potency of the plasma equal to 1 KW.

4.2.4 Hybrid Deep Model with State-Space Reduction

Considering a perfectly mixed fed-batch bioreactor, the macroscopic material balance equations take the following state-space form (Equation 35):

Equation 35. Perfectly mixed fed-batch bioreactor state-space equation

$$\frac{dC}{dt} = r + DC_{in} - DC$$

with C a $(m \times 1)$ vector of state variables (concentrations in the liquid phase), r a $(m \times 1)$ vector of volumetric reaction rates, $D = \frac{F}{V}$ the dilution rate, F the feed rate to the bioreactor, V the liquid volume inside the bioreactor, and C_{in} a $(m \times 1)$ vector of concentration in the feed stream to the bioreactor. The $m = 9$ concentrations included in the state vector, C, were those of biomass (X), recombinant protein (scFv), methanol (Met), ammonium ion (NH_4), Mg, K, Ca, P and S.

The $(m \times 1)$ vector of cumulative reacted amount of each compound at a given time t, $IR(t)$, is defined as the time integral of the respective reaction rates as follows (Equation 36):

Equation 36. Reacted mass definition.

$$IR(t) = \int_0^t r(\tau)V(\tau)d\tau$$

By combining Equation 35 and Equation 36, $IR(t)$ may be estimated from measured data of concentrations and culture volume (for simplicity we assume negligible sampling and bleeding volume) as follows (Equation 37):

Equation 37. Reacted mass estimation from measured concentrations and volume.

$$IR(t) = C(t)V(t) - C(0)V(0) - (V(t) - V(0))C_{in}$$

Using Equation 37, a transformed data matrix IR (with the same size as C) was computed for each fed-batch experiment with rows representing process time and columns the cumulative reacted amount of compounds (X, scFv, Met, NH4, Mg, K, Ca, P and S). The IR matrices of all fed-batch experiment were stacked vertically in a single matrix and then normalized by dividing each column by the respective absolute maximum value, IR_{max} (Equation 38):

Equation 38. Reacted mass normalization.

$$IR_{norm} = IR \oslash IR_{max}$$

with $\oslash$ the Hadamard division. The data matrix IR_{norm} was decomposed in a matrix of scores, S_{co}, and a matrix of coefficients, $Coeff_{norm}$, by PCA using the alternating least-squares algorithm (MATLAB function "pca" with option ALS) (Equation 39). This step was performed with the objective of data compression by choosing a number of principal components $NPCA < m$.

Equation 39. PCA decomposition of normalized reacted mass

$$IR_{norm} = S_{co} \times Coeff_{norm}^{T}$$

A denormalized form of Equation 39 was obtained by multiplying with IR_{max} (Equation 40):

Equation 40. Denormalized PCA decomposition

$$IR = S_{co} \times Coeff^{T}$$
$$Coeff = Coeff_{norm} \otimes IR_{max}$$

with $\otimes$ the Hadamard multiplication.

Recognizing that the IR is obtained by the time integral of reaction rates (Equation 36) then the compression of IR data according to Equation 40 with $NPCA < m$ has implicit a reduction of the volumetric reaction rates, r, to $NPCA$ linearly independent reaction rates, r_Z (Equation 41):

Equation 41. Correlation between the volumetric reaction rates and the NPCA reaction rates

$$r = Coeff \times r_Z$$

Equation 35 was finally transformed in a reduced state-space equation by replacing the volumetric reaction rates, r, in Equation 35 by Equation 41 and then by multiplying each term by the pseudo-inverse of $Coeff$, resulting in the following reduced state-space model (Equation 42):

Equation 42. Reduced state-space model equations

$$\frac{dZ}{dt} = r_Z + DZ_{in} - DZ$$
$$Z = pinv(Coeff) \times C$$
$$Z_{in} = pinv(Coeff) \times C_{in}$$

The reduced state space-model is then completed with the linear measurement model (Equation 43):

Equation 43. Linear measurement model

$$C = C_{oeff} \times Z$$

Furthermore, given that methanol feeding has a cumulative toxic effect in the metabolism of *P. pastoris*, an ODE that confers intracellular memory was added to the model (Equation 44):

Equation 44. Intracellular memory ODE

$$\frac{dSH}{dt} = -r_{SH} SH$$

with SH the shock factor with initial value $SH(0) = 1$ and r_{SH} the rate of variation of the shock factor. The shock factor is thus an internal unmeasured state variable. A similar ODE has been proposed by (Lee and Ramirez, 1992).

The reactions rates are described by a FFNN with nh hidden layers as follows (Equation 45):

Equation 45. FFNN model for the GS115 (Mut+) *Pichia pastoris* model with state-space reduction

$$H^0 = [Z \oslash Z_{max}, SH/SH_{max}, T/T_{max}, pH/pH_{max}]^T$$
$$H^i = \sigma\left(w^i \cdot H^{i-1} + b^i\right), \qquad i = 1, \dots, nh$$
$$[r_Z, r_{SH}]^T = w^{nh+1} \cdot H^{nh} + b^{nh+1}$$

The input layer $i = 0$ receives the information of the reduced state space vector, Z, internal state, SH, cultivation temperature, T and pH (Z_{max}, Sh_{max}, T_{max} and pH_{max} are the absolute maximum of Z, SH, T and pH respectively). Each hidden layer i computes a vector of outputs, H^i, from a vector of inputs, H^{i-1}, which are the outputs of the preceding layer. The transfer function of hidden nodes, $\sigma(.)$, was always the rectified linear unit *ReLU*. The output layer computes the reaction rates in the reduced reaction space. The parameters $w = \{w^1, w^2, \dots, w^{nh+1}\}$ are the nodes connections weights and $b = \{b^1, b^2, \dots, b^{nh+1}\}$ the bias

weights. The resulting hybrid model structure with state-space reduction is represented in Figure 10:

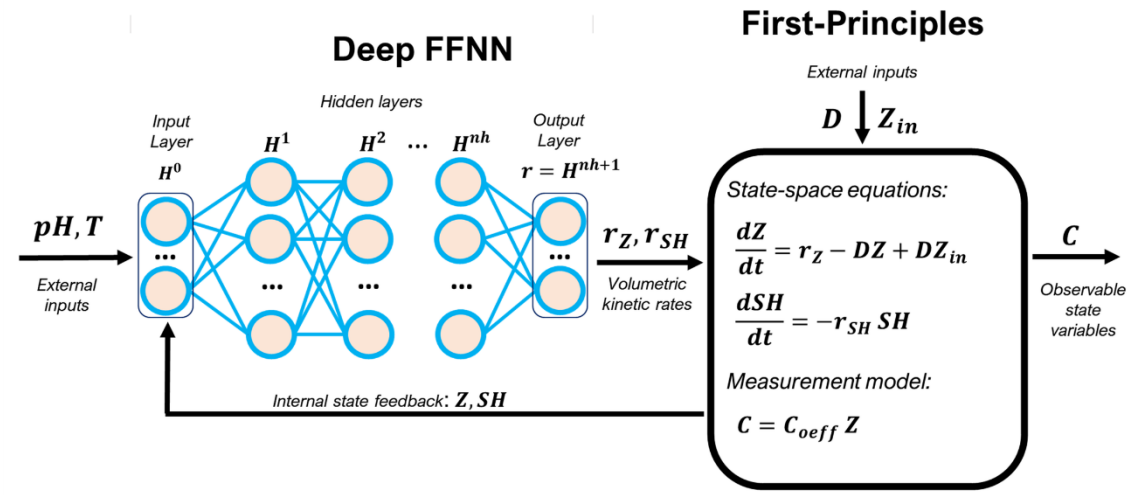

Figure 10. Hybrid model structure with state-space reduction for the methanol fed-batch phase (MFB) of the *P. pastoris* GS115 (Mut+) fed-batch process. The kinetic rates are defined nonparametrically by a deep FFNN. Bioreactor dynamics are defined parametrically by macroscopic material balance equations in a perfectly mixed vessel. The observable state variables are the concentrations of m=9 compounds, $C = [X, scFv, Met, NH4, Mg, K, Ca, P, S]^T$. The compressed internal state, Z, depends on the number of columns of the PCA coefficients matrix, $Coeff$. The PCA coefficients are obtained by unsupervised learning using data of cumulative reacted amount. The FFNNs weights are trained with a deep learning method based on ADAM, stochastic regularization and semidirect sensitivity equations as described by (Pinto et al. 2022).

All developed hybrid models were focused on the production MFB phase. The hybrid models were trained with the deep learning method proposed by (Pinto et al. 2022) based on the ADAM method adapted to dynamic hybrid models. Briefly, the MFB phase data were portioned in a training and a testing data subset (more details in section 4.3). The network weights were optimized on the training subset only (minimization of the weighted mean square error) using the ADAM algorithm and stochastic regularization. The objective function gradients were computed dynamically by the semidirect sensitivity equations method. For more details the reader is referred to (Pinto et al. 2022). Two different metrics were adopted to compare the hybrid models. The weighted mean square error (WMSE) was computed as (Equation 46):

Equation 46. Objective function for the reduced state-space models

$$WMSE = \frac{1}{T}\sum_{t=1}^{T}\frac{(c_t^* - c_t)^2}{\sigma_t^2}$$

with T the number of data points, c_t^* the observed concentration at time t, c_t the predicted concentration at time t and σ_t the standard deviation of measurement at time t. The WMSE

was minimized during the training and was also calculated for the test partition at the end of the training. The second metric was the Akaike Information Criterion with second order bias correction (AICc):

Equation 47. Akaike Information Criterion with second order bias correction

$$AICc = T\ ln(WMSE) + 2\ nw + \frac{2\ nw\ (\ nw\ +\ 1\ )}{T\ -\ nw\ -\ 1}$$

The AICc was computed on the training partition only and is used to compare hybrid models of different complexity (e.g. different number of network parameters, nw).

All the code was developed in-house and implemented in MATLAB on a computer with Intel(R) Core(TM) i5-8265U CPU @ 1.60 GHz 1.80 GHz, and 24 GB of RAM. The CPU time of the different tests performed were computed as the difference between the result of the "cputime" function in MATLAB. The source code and an example hybrid model implementation for the case study is accessible at: https://github.com/sbegroup-nova/HYBMOD.

4.3 Results and Discussion

4.3.1 Cultivation Experiments

Nine 50 L fed-batch cultivations were performed with varying pH, temperature and feeding profiles of glycerol and methanol in order to analyze the effect of reactor operational parameters on process dynamics. The temperature and pH were always the same in the GBFB phase (30°C and pH 5.0, respectively). In the MFB phase, the temperature levels were 23.6°C or 30°C whereas the pH levels were 4.0, 5.0, 6.5 or 7.0. Two experiments (A and E) were performed at baseline conditions (T=30°C and pH 5.0 according to Invitrogen guidelines). The overall results are summarized in Table 4. The final biomass concentration varied between 428.1±3.8 and 598.1±7.1 gWCW/L (40% variation) whereas the endpoint scFv titer varied almost tenfold (between 5.9±0.4 and 54.4±1.3 mg/L). As discussed below, the experiments A and F with the lowest and highest endpoint scFv titer were selected for testing while the remaining 7 experiments (B, C, D, E, G, H, and I) were selected for training the hybrid models. The product/biomass yield varied more than sixfold between 42 and 243.3 µg of scFv per unit of gWCW produced in the MFB phase. Experiment H, performed at 30°C and pH 6.5, resulted in the highest scFv yield (243.3 µg of scFv per unit of gWCW produced). Experiment I was performed at similar conditions to experiment H except for the higher pH 7.0. This experiment

delivered one of the lowest yields (45.7 µg of scFv per unit gWCW produced) denoting a very significant effect of pH on the process kinetics.

Table 4. Summary of 50 L fed-batch cultivation experiments performed and respective production yields. In the glycerol batch/ fed-batch phase (GBFB) the glycerol feeding program varied but the temperature and pH were 30°C and pH 5.0 in all cases. Temperature, pH, and methanol feeding in the methanol fed-batch phase (MFB) varied from experiment to experiment.

Exp.	Glycerol batch/fed-batch (GBFB)			Methanol fed-batch (MFB)						
	Δt (h)	Glycerol feed (kg)	Final X (gWCW/L)	Δt (h)	T(°C)	pH	Methanol feed (kg)	Final X (gWCW/L)	Final scFv (mg/L)	Yield scFv/X (µg/gWCW)
A	46.8	1.285	316.9±3.2	53.7	30.0	5.0	7.516	457.3±3.8	5.9±0.4	42.0
B	76.7	2.821	447.3±2.7	50.5	23.6/30.0*	5.0	9.540	585.0±0.5	15.6±2.2	113.3
C	47.3	1.264	295.4±1.1	98.0	23.6	5.0/7.0**	14.794	587.7±2.2	16.1±2.5	55.1
D	50.3	1.218	268.1±1.6	95.5	23.6	5.0/7.0***	19.002	573.1±1.1	14.3±1.8	46.9
E	53.4	1.285	301.5±2.7	70.5	30.0	5.0	13.338	434.2±3.8	11.9±1.3	89.7
F	48.3	0.586	164.2±6.0	136.7	23.6	4.0	23.602	598.1±7.1	54.4±1.3	125.4
G	48.0	1.031	274.2±10.3	102.0	23.6	4.0	9.808	479.6±1.6	30.7±0.6	149.5
H	47.3	1.037	259.6±10.3	105.5	30.0	6.5	12.189	475.4±3.3	52.5±8.6	243.3
I	46.0	1.034	244.2±22.3	103.0	30.0	7.0	10.488	428.1±3.8	8.4±0.5	45.7

(*) – transition occurred at t=121.2 h; (**) – transition occurred at t=123.0 h; (***) – transition occurred at t=125.0 h.

4.3.2 Inorganic Elements Dynamics

The dissolved concentration of inorganic elements Ca, Mg, K, S and P were assayed in the supernatant by ICP-AES during the MFB phase. Figure 11 shows the percentual variation of dissolved concentrations over time. These data show that in a typical BSM *P. pastoris* cultivation, run at 30°C and pH 5.0 according to Invitrogen guidelines (experiments A and E), Ca and S tend to be in excess whereas K, P and Mg tend to deplete sooner. In general, Mg tends to deplete first as seen in experiments C, D, H, and I. (Cos et al. 2006) reported the precipitation of BSM salts at pH higher than 5.0 and the same was observed in the present study (Figure 2). Precipitation occurred in experiments H and I, which were performed at pH 6.5 and 7.0 respectively during the MFB phase. The shift from pH 5.0 to 6.5 or 7.0 caused severe precipitation of Mg and Ca salts as evidenced by the sudden variation of the respective dissolved

concentrations, close to -100% in the case of Mg (e.g. complete depletion) and close to -80% in the case of Ca. The precipitation of other salts also occurred but not as severely. In experiments C and D, a pH shift from pH 5.0 to 7.0 occurred in the middle of MFB phase (at 123.0 h and 125.0 h respectively). This caused some precipitation of salts in both experiments. In the experiment that reached the highest bio-mass concentration (598.1 g-WCW.L-1, in experiment F), Mg and K depleted while P almost depleted. In the experiment that reached the lowest biomass concentration (428.1 g-WCW.L-1 in experiment I), salts precipitation occurred early in the culture, caused by the pH shift to 7.0 at the methanol induction point. Overall, these data suggest a strong correlation between pH, growth kinetics and salts precipitation. There seems to be a clear challenge to optimize the salts concentrations in the medium for high cell density *P. pastoris*. But even if the medium composition is optimized e.g. by statistical design of experiments, the salts concentrations will significantly decrease as cells grow over time. The salts dynamics may strongly affect the growth and protein expression kinetics in different phases of the process. Other factors such as temperature and methanol feeding rate may also play an important role. Understanding the combined dynamic effects of all Critical Process Parameters (CPPs) requires in-depth data analysis using a suitable dynamic modeling framework as discussed next.

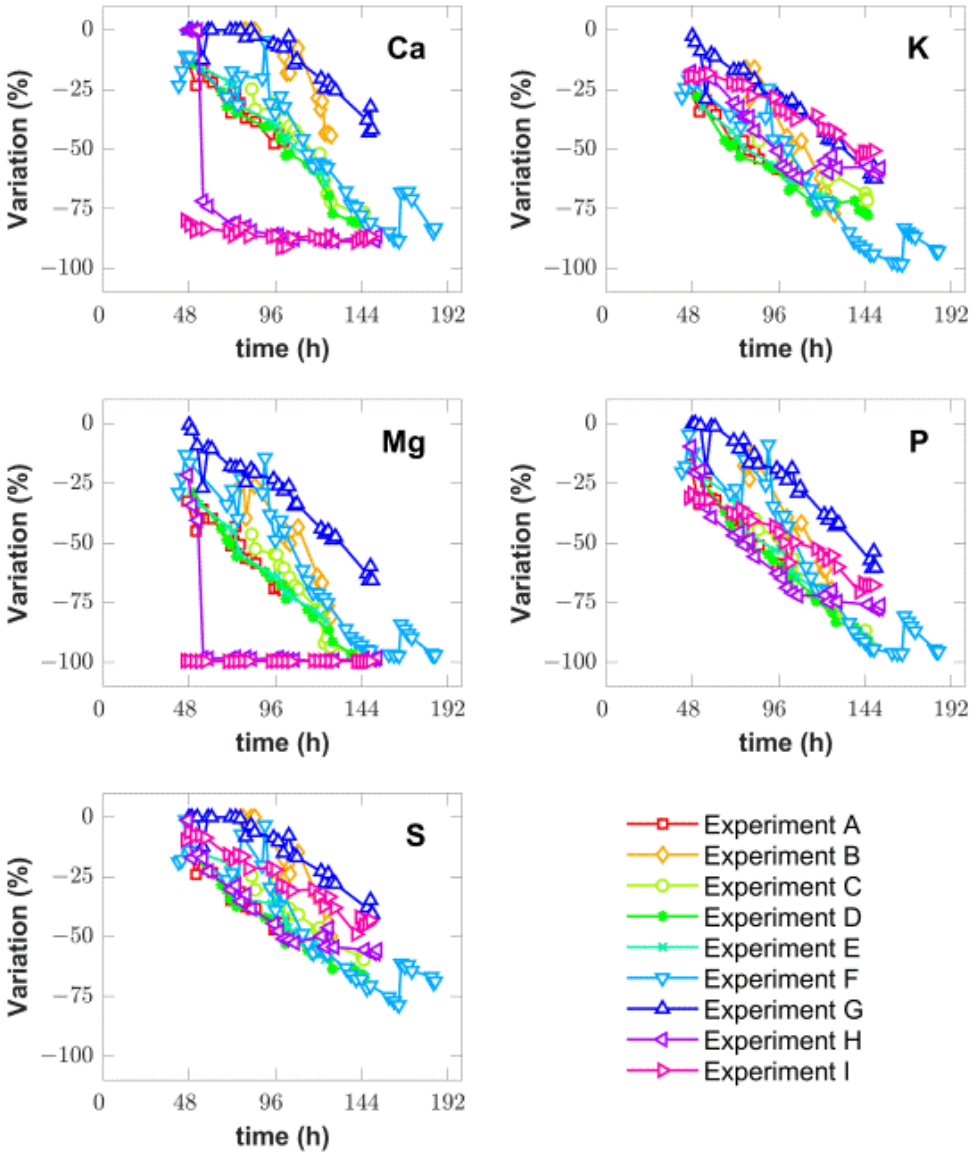

Figure 11. Percentual variation of dissolved inorganic elements concentrations (Ca, K, Mg, P and S) determined from ICP-AES measurements for each reactor experiment A-I (Table 4). The percentual variation of concentration was calculated as $\frac{c_i(t)-c_i(0)}{c_i(0)} \times 100$ with $c_i(t)$ the concentration of element i at cultivation time t.

4.3.3 PCA of Cumulative Reacted Amount

Data analysis started with the computation of the cumulative reacted amount over time, $IR(t)$, of the 9 bioreactor compounds (X, scFv, Met, NH4, Ca, K, Mg, P and S) for each experiment (A-I, Table 4) using the previously described method. In the case of Met and NH4, the variation of concentrations in the liquid were assumed to be negligible in comparison to the cumulative amount metabolized by the cells. In the case of inorganic elements, it was not

possible to distinguish between cellular uptake and precipitation/dissolution caused by the varying reactor conditions. The computed cumulative reacted amount thus aggregated both kinetic terms in the case of inorganic elements.

The IR data were normalized column wise by dividing with the maximum absolute value of each column. The normalized data was subject to PCA for a maximum number of principal components equal to 8 ($NPCA = 8$). The data were partitioned into 7 fed-batch experiments (B, C, D, E, G, H, and I) for the PCA and 2 experiments (A and F) for validation. The validation experiments corresponded to the extreme low and high scFv endpoint titer experiments. The overall results are shown in Figure 3. The resulting 9×8 coefficients matrix (Equation 48), with rows representing bioreactor compounds and columns the PCs, was used in the state-space reduction step described in the following section.

Equation 48. PCA coefficients matrix

$$Coeff_{norm} = \begin{bmatrix} 0.32 & 0.38 & 0.13 & 0.03 & 0.05 & 0.35 & 0.65 & 0.43 \\ 0.21 & 0.02 & 0.22 & 0{,}68 & -0{,}60 & 0.13 & -0.24 & 0.06 \\ -0.42 & 0.63 & -0.17 & -0.19 & -0.20 & 0.45 & -0.32 & -0.04 \\ -0.19 & -0.16 & 0.85 & -0.12 & 0.24 & 0.34 & -0.14 & -0.04 \\ -0.55 & 0.33 & 0.09 & 0.50 & -0.28 & -0.35 & 0.23 & -0.04 \\ -0.32 & -0.09 & 0.18 & -0.10 & -0.25 & -0.37 & 0.01 & 0.60 \\ -0.11 & 0.08 & 0.22 & -0.43 & -0.61 & -0.25 & 0.34 & -0.27 \\ -0.34 & -0.37 & -0.14 & 0.21 & -0.14 & 0.36 & 0.46 & -0.38 \\ -0.33 & -0.41 & -0.27 & -0.04 & -0.09 & 0.31 & -0.02 & 0.49 \end{bmatrix}$$

Figure 12A shows that 2 to 4 principal components (PC) cumulatively explain 90.3%, 94.5% and 97.0% of data variance. These results evidence strong linear dependencies between the biochemical transformations involving the 9 bioreactor compounds. The PCA coefficients shown in Figure 12B (blue dots and blue lines) suggest a very strong correlation between biomass production, methanol consumption, NH4 consumption and K consumption along the directions of PC-1 and PC-2, which together explain 90.3% of data variance. The PC-1 and PC-2 are mainly associated with cell growth metabolic processes (all other PCs have low biomass coefficients) with PC-2 showing a minor contribution to scFv synthesis (low scFv coefficient). The scFv synthesis is mainly explained by PC-1, PC-3, and PC-4. The scFv synthesis appears positively correlated with cell growth along the direction of PC-1 (Figure 12B). However, in the biplots of PC-3 (4.2% explained variance, Figure 12C) and 4 (2.5% explained variance, Figure 12D), the coefficients of scFv and biomass are large and negligible, respectively, suggesting cell growth dissociated product synthesis. The interpretation of the inorganic elements coefficients is more difficult due to the occurrence of precipitation. The coefficients of PC-1 (75.6% explained variance) suggest that all inorganic elements are consumed for cell

growth with S showing the least significant contribution. The elements Ca and Mg, which precipitated more severely, are orthogonal to X along the direction of PC-1 and PC-2 (Figure 12B) suggesting a low correlation with biomass growth. The S also appears orthogonal to X denoting a low correlation with biomass growth. Biomass growth seems to be controlled mainly by Met, NH4 and K availability and to be practically insensitive to Ca, Mg and S availability. As for the product synthesis, the main contributions are from PC-1 (75.6% explained variance) and PC-3 (4.2% explained variance). In the case of PC-1 the conclusions already taken for biomass growth hold for scFv synthesis. As for the PC-3, the coefficients show again a low correlation with the Ca and Mg (direction of PC3 in Figure 12C). On the other hand, scFv appears positively correlated with K, S and P along the direction of PC -3 suggesting that excessive consumption of these elements (e.g. lower reaction rates) is associated with a lower scFv synthesis rate. This interpretation is only qualitative as the different PCs collectively contribute to explain data variance.

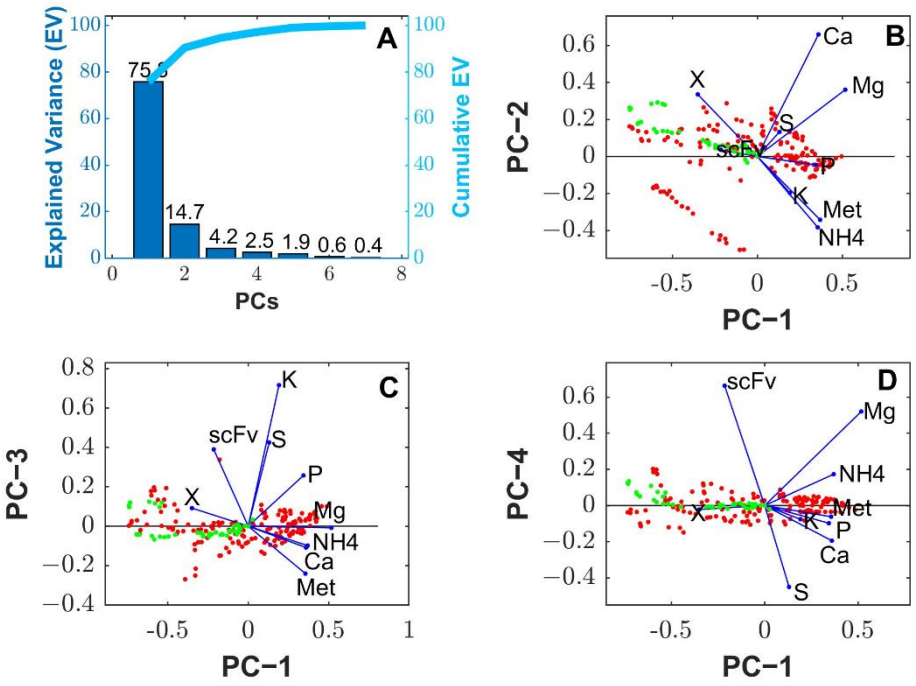

Figure 12. PCA of normalized cumulative reacted amount data of X, Met, NH4, Ca, K, Mg, P and S for 9 fed-batch experiments. Each column was divided by the maximum absolute value of reacted amount among the 9 fed-batch experiments. The PCA algorithm was the alternating least-squares (MATLAB function "pca" with option ALS). Red points are scores of training data. Green points are scores of validation data. Blue dots and blue lines are coefficients. **A**: explained variance over number of principal components. **B**: scores and coefficients of principal component 2 over principal component 2. **C**: scores and coefficients of principal component 3 over principal component 1. **D**: scores and coefficients of principal component 4 over principal component 1.

4.3.4 Hybrid Model Development

For a quantitative analysis of all critical process parameters (CPPs), hybrid models were developed to describe process dynamics using the previously described state-space reduction method. The PCA coefficients matrix obtained in the previous section was used to transform the concentrations vector in a reduced Z state-vector by applying the transformations (Equation 42).

Firstly, the effect of the state-space reduction on the hybrid model training and testing was investigated. Different hybrid models were developed by considering an increasing number of PCs, e.g. by taking an increasing number of columns of matrix Coeff. The same data partitioning as for the PCA was adopted, namely 7 experiments were selected for training (B, C, D, E, G, H, and I) and 2 experiments were selected for testing (A and F). The number of hidden layers was 2 with 10 nodes each, which were kept the same in all tests performed. The training method was also the same in all tests performed (ADAM with 1000 iterations and default

hyperparameters, stochastic regularization with 80% minibatch size and 20% weights dropout and semidirect sensitivity equations). The overall results are presented in Table 5. The final training error systematically decreased with the number of PCs. The lowest training error was achieved with the original unreduced state-vector of concentrations. However, a clear minimum in the test error is obtained for 5-6 PCs, which corresponds to a reduction of 40% and 30% in the number of state variables (6 and 7 respectively). The number of FFNN weights increased with the number of PCs but the AICc criterion failed to discriminate the model with the highest predictive power, which was the model with 5 PCs reduction.

Table 5. Effect of state-space reduction on the hybrid modeling results. The number of principal components was increased from 1 to 8 in the state space transformation defined by Equation 42. Seven fed-batch experiments were selected for training (B, C, D, E, G, H, and I) and 2 for testing (A and F). The training was performed with the ADAM algorithm with 1000 iterations and hyperparameters α=0.001, β1=0.9 β2=0.999 and $\eta=1\times10^{-7}$. Gradients were computed by the semidirect sensitivity equations. Stochastic regularization was applied with weights dropout of 0.2 and minibatch size of 0.8. The training was repeated only once with random weights initialization from the uniform distribution between -0.01 and 0.01.

Number of principal components	WMSE train	WMSE test	AICc	CPU time (hh:mm:ss)	Number of weights	Cumulative explained variance (%)
1	11.31	12.4	4380	02:19:00	182	72.25
2	3.45	4.47	2490	02:25:00	203	89.94
3	2.61	3.99	2090	02:20:00	224	95.24
4	0.98	1.97	550	02:30:00	245	97.77
5	**0.59**	**1.18**	**-300**	**02:24:00**	**266**	**98.85**
6	0.50	1.21	-430	02:22:00	287	99.33
7	0.37	1.40	-820	02:25:00	308	99.72
8	0.32	1.42	-1110	02:20:00	329	99.91
unreduced	**0.30**	**1.42**	**-1100**	**02:24:00**	**350**	**100.00**

For both the unreduced and 5 PCs reduced hybrid models, it was further investigated the optimal size of the FFNN. Several architectures were investigated with 1 to 3 hidden layers and with varying number of nodes in the hidden layers. The same training/testing data partitioning and training methods were adopted. Table 6 shows the results for the hybrid model with 5 PCs reduction. The best shallow hybrid structure had a single hidden layer with 13 nodes and 201 parameters. The training and testing errors were 0.50 and 1.10 respectively. The best deep structure had 2 hidden layers with 13 nodes each and 383 parameters. The

final training error was the same as the shallow structure (0.50), but the test error was slightly lower (1.01). Nevertheless, given the lower model complexity reflected in the lower AICc value, the shallow structure with 13 hidden nodes was taken as the best hybrid model with 5 PCs reduction. Table 7 presents the results for the unreduced hybrid model with varying FFNN sizes. The hybrid structure with 2 hidden layers 15×15 and 595 parameters stands out as the best model. It had a low training error (0.33), the lowest test error (1.35) and the lowest AICc value (-150).

Table 6. Hybrid modeling results as a function of FFNN size for 5 principal components reduction (6 state variables). Seven fed-batch experiments were used to train the model (B, C, D, E, G, H, and I) and 2 experiments were used for testing (A and F). The training was performed with the ADAM algorithm with 1000 iterations and hyperparameters α=0.001, β1=0.9 β2=0.999 and η=1e-7. Gradients were computed by the semidirect sensitivity equations. Stochastic regularization was applied with weights dropout of 0.2 and minibatch size of 0.8. The training was performed only once with random weights initialization from the uniform distribution between -0.01 and 0.01.

Number of hidden nodes	WMSE train	WMSE test	AICc	CPU time (hh:mm:ss)	Number of weights
5	1.57	3.19	910	02:10:00	81
6	0.95	2.11	170	02:14:00	96
7	0.89	1.88	56	02:12:00	111
8	0.67	1.54	-380	02:15:00	126
9	0.65	1.47	-390	02:16:00	141
10	0.57	1.26	-590	02:08:00	156
11	0.58	1.26	-520	02:26:00	171
12	0.57	1.27	-490	02:18:00	186
13	**0.50**	**1.10**	**-680**	**02:25:00**	**201**
14	0.52	1.31	-560	02:12:00	216
15	0.51	1.12	-540	02:13:00	231
[5 5]	1.05	2.08	320	02:05:00	111
[6 6]	0.80	1.76	-70	02:21:00	138
[7 7]	0.79	1.64	-10	02:28:00	167
[8 8]	0.63	1.31	-300	02:27:00	198
[9 9]	0.62	1.22	-230	02:33:00	231
[10 10]	0.59	1.18	-300	02:24:00	266

[11 11]	0.52	1.16	-310	02:32:00	303
[12 12]	0.58	1.21	-10	02:30:00	342
[13 13]	**0.50**	**1.01**	**-470**	**02:33:00**	**383**
[14 14]	0.58	1.22	250	02:40:00	426
[15 15]	0.59	1.23	450	02:32:00	471
[5 5 5]	0.94	2.10	220	02:24:00	141
[6 6 6]	0.76	1.76	-40	02:32:00	180
[7 7 7]	0.63	1.35	-230	02:28:00	223
[8 8 8]	0.69	1.41	50	02:39:00	270
[9 9 9]	0.59	1.22	-50	02:34:00	321
[10 10 10]	0.61	1.13	240	02:36:00	376
[11 11 11]	0.64	1.17	710	02:36:00	435
[12 12 12]	0.65	1.21	1170	02:39:00	498

Table 7. Hybrid modeling results as a function of FFNN size for the unreduced case (10 state variables). Seven fed-batch experiments were used to train the model (B, C, D, E, G, H, and I) and 2 experiments were used for testing (A and F). The training was performed with the ADAM algorithm with 1000 iterations and hyperparameters α=0.001, β1=0.9 β2=0.999 and η=1e-7. Gradients were computed by the semidirect sensitivity equations. Stochastic regularization was applied with weights dropout of 0.2 and minibatch size of 0.8. The training was performed only once with random weights initialization from the uniform distribution between -0.01 and 0.01.

Number of hidden nodes	WMSE train	WMSE test	AICc	CPU time (hh:mm:ss)	Number of weights
10	1.41	2.58	1080	02:15:00	240
15	0.48	1.87	-300	02:29:00	355
20	0.52	1.74	-120	02:30:00	470
[10 10]	0.30	1.42	-1100	02:28:00	350
[15 15]	**0.33**	**1.35**	**-1150**	**02:38:00**	**595**
[20 20]	0.42	1.38	210	02:44:00	890
[10 10 10]	0.41	1.48	-360	02:26:00	460
[15 15 15]	0.42	1.54	140	02:41:00	835
[20 20 20]	0.35	1.64	360	02:48:00	1310

Comparing the best reduced model (with five PCs' reduction and a single hidden layer with 13 nodes, Table 6) and the best unreduced model (two hidden layers with 15 nodes each, Table 7), it becomes clear that the state-space reduction had a very positive impact in the hybrid model performance metrics. The model complexity was reduced by 66% partially due to the lower number of state variables, which reflected in a lower number of FFNN inputs and outputs. Moreover, 1 hidden layer was removed comparatively to the best deep model. It may be argued that the PCA coefficients in Equation 41 act as a linear layer obtained by unsupervised learning (namely by PCA) downstream of the FFNN. Such structural differences resulted in a higher training error (51.5% higher) for the reduced shallow hybrid model but, more importantly, in a significantly lower testing error (18.5% lower). The AICc is not a good discrimination metric in this case because the training error is systematically lower for unreduced models thus always favoring unreduced structures. The reduced hybrid model predictions and respective measured concentrations of biomass, scFv and inorganic elements (Mg, K, Ca, P and S) for the two test experiments A and F in Figure 13. The model was able to faithfully predict the state variables for the two extreme experiments with predictions always within or very close to measurement error bounds.

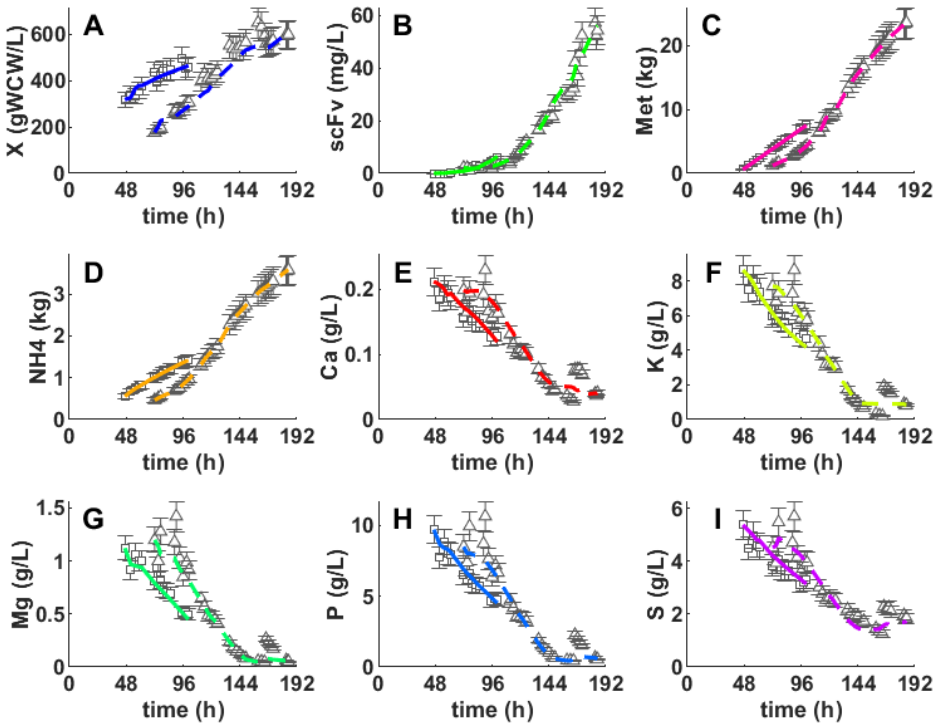

Figure 13. Comparison between predictions of the hybrid model 5x13x5 with 5 state variables and experimental data of X, scFv, Met, NH3, Ca, K, Mg, P and S for the 2 fed-batch experiments A and F. Squares and triangles are

measurements of experiment A and F respectively. Full line and dashed line are hybrid model predictions of experiment A and F respectively. **A**: biomass. **B**: single-chain antibody fragment(scFv). **C**: cumulative methanol consumption (kg). **D**: cumulative NH4 consumption (kg). **E**: calcium (Ca, g/L). **F**: potassium (K, g/L). **G**: magnesium (Mg, g/L). **H**: Phosphorus (P, g/L). **I**: sulfur (S, g/L).

4.3.5 Design Space Exploratory Analysis

Here we illustrate how the hybrid model can be used as a DT prototype for dynamic design space exploration. The CPPs that affect recombinant protein production by methylotrophic *P. pastoris* are typically the pH, temperature and methanol feeding strategy (e.g. (Jahic et al. 2003), (Jahic et al. 2006), (Vanz et al. 2012), Ferreira et al. (2014), (Looser et al. 2015)). In this study, the feeding of inorganic elements is also analyzed. The impact of CPPs on cell growth and scFv synthesis dynamics was characterized by process simulations using the best hybrid structure with 5 PCs reduction and 13 hidden nodes developed in the previous section (Table 6 and Figure 13). A sensitivity analysis was performed taking as reference condition the experiment H, which delivered the highest scFv/biomass yield. Thus, the objective is to analyze the feasibility of increasing the scFv yield beyond the value obtained in experiment H by optimizing CPPs.

The optimal pH and temperature in the production phase depend on the nature and function of the expressed protein and on the genetic modification of the host cells. Figure 14 shows a sensitivity analysis of scFv endpoint titer to temperature and pH for the recombinant strain used in this study. The inner rectangle represents the domain of experience covered by the 9 fed-bath experiments. These data suggest an optimal pH 5.75-6.75 and temperature 27.5-35°C region corresponding to a higher endpoint scFv titer. These results are aligned with the data reported by Joseph et al. (2022), obtained with a *P. pastoris* GS115 (Mut+) strain expressing recombinant thaumatin II. The authors consistently observed a higher viable cell density and higher secretion of protein at pH 6.0 compared to pH 5.0 (when the cells were grown at 30 °C) in different culture media. A low pH between 4.0-5.0 has been reported to decrease the proteolytic activity of proteases in the supernatant (Jahic et al., 2003). On the other hand, a high pH may counteract by increasing cellular viability thereby reducing the cell lysis and the release of proteases (Joseph, Akkermans, and Van Impe 2022). A trade-off between both mechanisms must be evaluated on a case-by-case basis. Protein folding may also be severely affected by temperature. Misfolded proteins can lead to a higher degradation rate in the cytosol and ultimately to a lower secretion rate. Joseph et al. (2022) observed that

the protein levels were the highest at 30 °C compared to 20 and 25 °C at pH 6.0 thus the decrease of temperature did not improve the final titer. These results are in line with the optimal PH-temperature space identified in the present study. The identified optimal region encompasses experiment H (conducted at 30°C and pH 6.5), which delivered the highest scFv/biomass yield of 243.3 µg/gWCW. It may be thus concluded that, for the strain used in the present study, temperature and pH optimization have low potential for further scFv titer improvement.

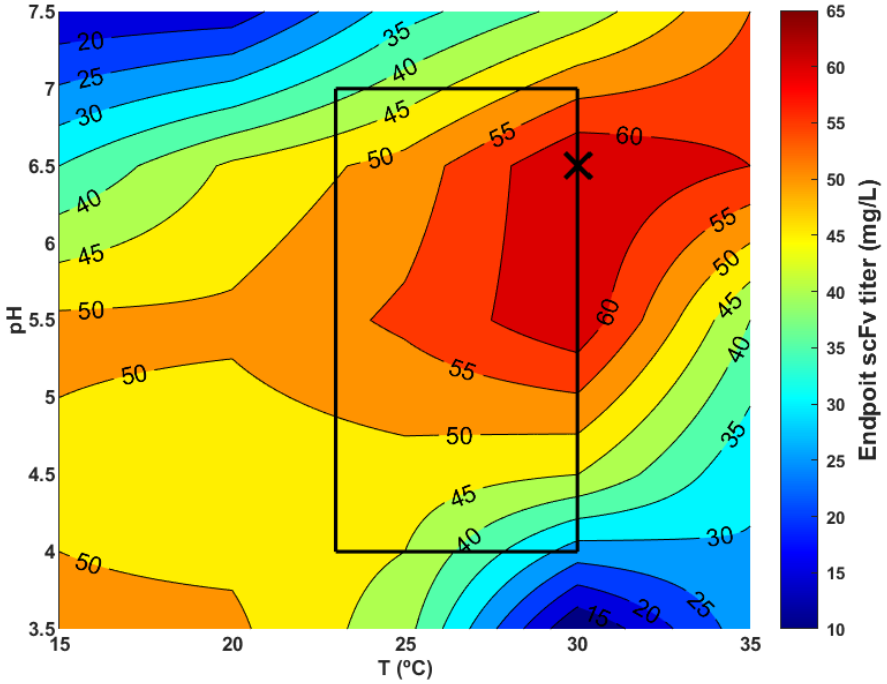

Figure 14. Sensitivity analysis of scFv endpoint titer to temperature (15-35 °C) and pH (3.5 – 7.5). The methanol feeding strategy was that of experiment H (reference condition). Data was obtained by simulations of the hybrid shallow model with 5 PCs reduction. The inner square represents the domain of experience. The cross marker represents the temperature (30°C) and pH (6.5) conditions of experiment H (reference condition).

The methanol feeding rate also plays a critical role in the *P. pastoris* GS115 Mut+ ex-pression system. The protein expression is controlled by the very strong AOX1 promoter induced by methanol. Methanol also serves as the main carbon source for cell growth and protein expression. Overflow methanol metabolism may lead to the accumulation of reactive oxygen species and a pronounced oxidative stress response (Vanz et al., 2012). Protein expression kinetics in the Mut+ P. pastoris phenotype may vary considerably from strain to strain. It may be growth coupled, negative growth related and bell shaped in relation to the specific growth rate profile (Looser et al., 2015). The methanol feeding rate is typically used to control the specific growth rate and the associated specific protein expression rate. This control

needs to be optimized on a case-by-case basis. Figure 15 shows a sensitivity analysis of scFv endpoint titer to the methanol feeding strategy for the strain used in this study. Again, the most productive experiment H served as a reference condition. The pH was varied between 3.5 and 7.5. The temperature was kept constant at 30°C. The methanol feeding was decreased or increased in relation to the experiment H feeding program by a multiplying factor between 0.25 and 1.5. The overall results show that there is a significant potential for scFv endpoint titer improvement by increasing the methanol feeding rate. Specifically, the pH region between 5.5-6.5 combined with 25% methanol feed rate in-crease (in relation to experiment H) has a scFv endpoint titer improvement potential of about 30%.

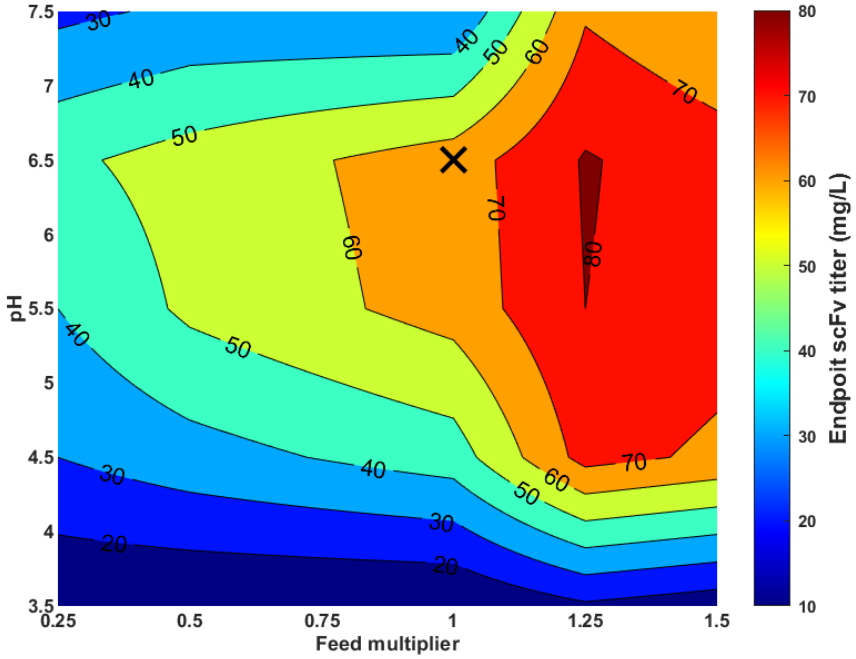

Figure 15. Sensitivity analysis of scFv endpoint titer to pH (3.5 – 7.5) and methanol feeding program (0.25 to 1.5 multiplying factor in relation to the methanol feed program applied to experiment H). The temperature was kept constant at 30°C. The data was obtained by simulations of the hybrid shallow model with 5 PCs reduction. The cross marker represents the reference condition of experiment H.

The high concentration of salts in the BSM medium is required to supply inorganic elements at sufficient stoichiometric quantities to sustain high cell density. An indication of this are the PC 1 coefficients (First column of Equation 48 and biplot of Figure 12B) showing that all inorganic elements have a significant contribution to the production of biomass. A common problem is however precipitation (Figure 11). The dilution of BSM medium to one-quarter has been studied by Brady et al. (2001) to mitigate the precipitation problem. The authors utilized a low salt medium that did not reduce growth rates nor protein expression rates while avoid-

ing medium precipitation. They observed no adverse effect on both glycerol and methanol growth kinetics. Later on, the dilution of BSM was shown to increase *P. pastoris* cellular viability and to reduce the cell death rate (Surribas et al. 2007) (Zhao et al. 2008). The reduction of the cell death rate decreases the accumulation of proteases in the supernatant and therefore the proteolytic attack on the secreted protein. Furthermore, the excess of trace metals was shown to decrease the expression of β-galactosidase by P. pastoris GS115 (Mut+) (Plantz et al. 2007). More recently, Joseph et al. (2022) compared different media and concluded that BSM resulted in the highest total cell concentration (as measured by dry cell weight) concomitantly with the lowest viable cell concentration. The high concentrations of salts may cause high osmotic stress to the cells resulting in a decrease of metabolic efficiency, cellular viability and in an increase of the cell death rate (Zhao et al. 2008). The higher cell death rate causes the release of proteolytic enzymes to the medium and a higher degradation of the expressed protein in the supernatant.

To test these hypotheses a set of dynamic simulations were performed with the hybrid model with 5 PCs reduction. The overall results are shown in Figure 16. Taking as reference the best experiment H (methanol feed program, 30°C and pH 6.5), one simulation was performed with a reduction of inorganic elements concentrations at the onset of the MFB phase to one-quarter. Another simulation was performed with controlled inorganic elements concentrations to constant values corresponding to one-quarter of BSM concentrations throughout the complete MFB phase.

The simulation with reduction to one-quarter of the initial salts concentrations showed no significant effect in the beginning of the MFB phase until approximately 72 hours. This is in accordance with the experimental results reported by (Brady et al. 2001). After 72 hours of cultivation, severe cell growth limitation by inorganic elements is forecasted. The much lower cellular concentration resulted in a significant reduction of the scFv titer. This simulation suggests that a BSM/4 diluted medium is no longer able to sustain high cell density.

The second simulation with inorganic elements control to constant levels suggests a very significant increase in the scFv endpoint titer by 80% in relation to the reference condition and also an increase in the final biomass by approximately 15%. The cell growth rate decreased but the growth phase was extended to a longer period of time. The scFv specific productivity was boosted by keeping the salts at a constant and low concentrations level. These results are in accordance with the experimental data reported by (Jahic et al. 2006). The authors developed a salts control system based on on-line conductivity monitoring in a *P.*

pastoris process. The control of conductivity at 8 mS.cm-1 resulted in a 3.6fold titer increase in relation to a standard BSM cultivation.

Overall, the design space analysis suggests that the control of inorganic salts in the MFB phase has the highest potential to further increase the scFv yield for the recombinant *P. pastoris* strain under study.

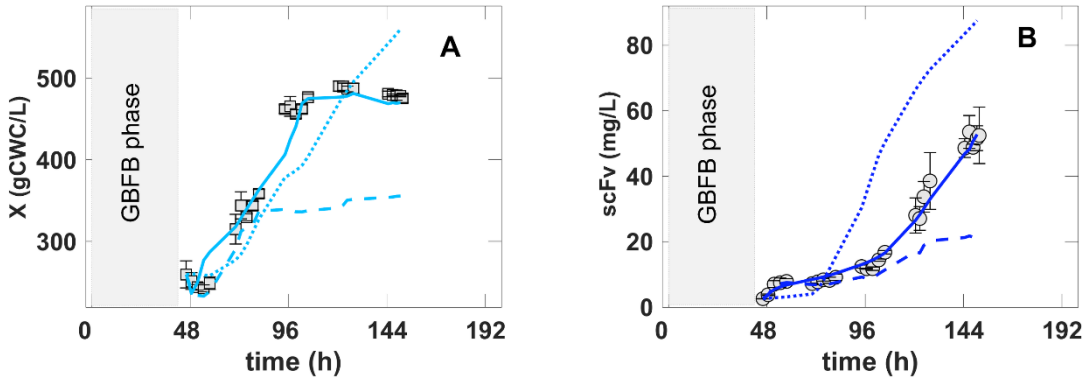

Figure 16. In silico experiments obtained by simulations of the hybrid shallow model with 5 PCs reduction based on experiment H control degrees of freedom. Symbols and error bars are measured data points of reference condition (experiment H). Full line is the hybrid model simulation of reference condition (experiment H). Dashed line is the hybrid model simulation of reference condition with one-quarter reduction of initial concentrations of inorganic elements at the onset of the MFB phase. Dotted line is the hybrid model simulation of reference condition with inorganic elements concentrations controlled to constant values corresponding to one-quarter of BSM concentrations throughout the complete MFB phase. **A** – biomass concentration over time. **B** – scFv titer over time.

4.4 Conclusions

In this chapter the dynamics of the main inorganic elements in *P. pastoris* GS115 (Mut+) cultures expressing a scFv were investigated. The ICP-AES data showed excess of Ca and S over Mg, P and K in BSM medium. In some cultures, Mg, P and K depleted completely eventually limiting biomass growth and scFv expression. Precipitation occurred during the MFB phase at pH 6.5 and 7.0, more severely for Ca and Mg. A hybrid modeling framework with state-space reduction was applied for data analysis and design space exploration. The state-space reduction framework succeeded in decreasing the model complexity by 60% and improving the predictive power by 18.5% in relation to a standard nonreduced hybrid model. The reduced hybrid model was able to correctly simulate the experiments performed including the test experiments. However, more data is required to strengthen model validation before it can be considered for a process digital twin. An exploratory sensitivity analysis of process dynamics to CPPs was performed. It was concluded that a temperature of 30 °C and pH 6.5 are close to the optimal operating point. Interestingly, at these conditions the culture suffered from se-

vere salts precipitation resulting in the highest scFv/biomass yield. The methanol feeding sensitivity analysis showed a significant 30% scFv endpoint titer improvement potential. The optimization of the inorganic elements feeding showed the highest potential for further scFv endpoint titer improvement. Namely, the control of inorganic elements concentration to one-quarter of the BSM during the MFB phase displayed an 80% scFv endpoint titer improvement potential.

5

Hybrid Deep Modeling of a CHO-K1 Fed-Batch Process

This chapter is based on the publication: Pinto, J., Ramos, J. R., Costa, R. S., Rossell, S., Dumas, P., & Oliveira, R. (2023). Hybrid deep modeling of a CHO-K1 fed-batch process: combining first-principles with deep neural networks. *Frontiers in Bioengineering and Biotechnology*, *11*.

5.1 Introduction

Chinese hamster ovary (CHO) cells are the most widely used host system for the industrial production of biologics. They cover more than 70% of the mammalian cell-based therapeutic proteins production (Vcelar et al., 2018). They present several advantages such as well-established large-scale cultivation with high productivity (cell densities higher than 20 Mcell/mL with protein titer as high as 10 g/L), human-like N-glycosylation, well-established molecular biology techniques and an impressive track record of approvals by the U.S. Food and Drug Administration (FDA) (Galleguillos et al., 2017). Given its industrial relevance, many companies have established CHO-cell platforms to streamline process development of many different molecule candidates in a short timeframe (e.g., (Mora et al., 2018)). Different up-stream tasks such as clone screening, culture media customization and reactor optimization should be integrated in a rational way to improve the efficiency of process development. The adoption of high-throughput screening technologies allied with advanced digitalization tools for data analysis, mathematical modeling and control across the different development stages are key factors to improve process development efficiency (Hole et al., 2021).

There are currently three main mathematical modeling formalisms that are used for the digitalization of biopharmaceutical processes: First-Principles or mechanistic modeling (e.g.,

Hartman et al., 2022, Monteiro et al., 2023), data-based or machine learning (ML) (e.g., Monbray et al., 2022, Helleckes et al., 2023) and hybrid mechanistic/ML (e.g., Badr and Sugiyama, 2020, Narayanan et al., 2023, Bayer et al., 2023). Mechanistic modeling relies on prior process knowledge and requires less process data. Conversely, ML relies almost exclusively on process data with minimal prior knowledge requirements. Mechanistic models are more complex to develop but tend to extrapolate better outside the domain of experience. The intrinsic complexity of biological systems is however a critical limitation for the deployment of mechanistic models in an industrial context (Badr et al., 2021). Data-driven and ML methods are easier to develop but require large amounts of data that are costly, time-consuming, and difficult to reuse. ML models tend to describe better inside the domain of experience (e.g., better interpolation) but are less reliable at extrapolating in comparison to mechanistic models. Hybrid models combine mechanistic and ML techniques in a common workflow and share the pros and cons of both techniques (e.g., Psichogios and Ungar, 1992, Oliveira, 2004, Teixeira et al., 2005, Teixeira et al., 2007, von Stosch et al., 2014, Kurz et al., 2022, Pinto et al., 2019). The mechanistic modules allow to decrease the complexity of the ML modules within the hybrid model and as such the overall data requirements are decreased. Moreover, the ML modules fill the gaps of the mechanistic modules for which knowledge is still lacking. Narayanan et al. (2022) studied the impact of increasing the amount of prior knowledge (e.g., material balances, reaction stoichiometry and reaction kinetics) in the hybrid model of a cell culture process. Between a fully data-driven (or ML model) and a fully mechanistic model, there are different degrees of hybridization possible depending on the amount of prior knowledge included in the hybrid model. The authors concluded that the inclusion of unbiased prior knowledge progressively improves the performance of the hybrid model. Unsurprisingly, fully data-driven models showed poor performance particularly when data is scarce. Rogers et al (2023) have also investigated the optimal amount of prior knowledge to incorporate in a hybrid bioprocess model. The authors concluded that the inclusion of correct kinetic information generally improves the performance of the hybrid. The inclusion of incorrect kinetic assumptions may however create inductive bias that decreases the performance of the hybrid model. Due to the flexible trade-off between prior knowledge and data availability, hybrid modeling is becoming a method of choice to develop digital twins in the realm of Biopharma 4.0 (e.g. Badr and Sugiyama, 2020, Yang et al., 2019, Sansana et al., 2021, Sokolov et al., 2021, Badr and Sugiyama, 2020, Narayanan et al., 2023, Bayer et al. (2022), Bayer et al. (2023)).

Being the preferred host system in biopharma, CHO cultivation processes have been the object of several hybrid modeling studies (Table 8). Most of the previous studies combined macroscopic material balance equations of extracellular species with some machine learning/statistical modeling methods with predominance of shallow FFNNs with a single hidden layer. The macroscopic material balance equations are translated to systems of Ordinary Differential Equations (ODEs) describing bioreactor dynamics. The machine learning component is typically dedicated to model biological kinetics, which are parts of the system lacking mechanistic basis. The number of biochemical species has been limited to 2-12 species. Typically, the viable cell count, concentrations of the target molecule and the concentrations of key central carbon metabolites such as glucose, lactate, glutamine, glutamate, and ammonium. A recent study by Doyle et al. (2023) has also covered amino acids dynamics. The training method is either coupled or uncoupled. In the latter case, the machine learning component is isolated from the mechanistic model and trained as a standalone module. In the former case, the mechanistic and machine learning models are parametrized in a common mathematical structure and trained together. Uncoupled training has been adopted by Kotidis et al. (2021) to develop a hybrid model of glycosylation critical quality attributes in CHO cultures. The N-linked glycosylation was described by a FFNN with 2 hidden layers, while the cell growth and metabolism were described by a mechanistic model based on a system of Differential and Algebraic Equations (DAEs) (Kotidis et al., 2019). The FFNN was trained as a standalone model on data generated by the mechanistic model using the TensorFlow package in Python 3.7. The final trained FFNN and the mechanistic model were assembled in a hybrid workflow in gPROMS v.5.0.1. Coupled training has been the preferred approach for material balance + FFNN hybrid models, following the scheme originally proposed by Psichogios and Ungar (1992). The sum of square error between measured and calculated concentrations is minimized during the training using the Levenberg-Marquardt (LMM) algorithm. Since the FFNN outputs cannot be directly compared with measured properties, this method is termed indirect training. The indirect sensitivity equations are employed to compute the gradients of measured concentrations in relation to neural network weights (Psichogios and Ungar, 1992, Oliveira, 2004). Cross-validation techniques are employed to avoid overfitting. Following the coupled training approach with cross-validation, Bayer et al. (2022) compared mechanistic and shallow hybrid modeling for characterization of a CHO cultivation process. The authors concluded that the prediction accuracy of the shallow hybrid model was always superior to the mechanistic model irrespective of the utilized data partition. Due to its' higher fitting power, the shallow hybrid model prediction accuracy showed to be more sensitive to data

resampling than the mechanistic model. Every hybrid model in Table 8 is of dynamic nature except the one by Ramos et al. (2022). The authors have used a large genome-scale network with 788 reactions as mechanistic component combined with a Principal Component Analysis (PCA) model. The overall hybrid model is of static nature, solved by linear programming under the pseudo steady-state hypothesis, i.e. by hybrid Flux Balance Analysis (hybrid FBA).

Table 8. Compilation of CHO hybrid modeling studies

First-Principles	Machine learning	Training method	Cross validation	Objective	Reference
Macroscopic material balances (2 species)	Shallow FFNN (tanh hidden nodes)	Levenberg-Marquardt; coupled	Yes	Prediction of culture dynamics; Quality-By-Design	Bayer et al. (2021)
Macroscopic material balances (7 species)	Shallow FFNN (tanh hidden nodes)	Levenberg-Marquardt; coupled	Yes	Prediction of culture dynamics; Quality-By-Design	Bayer et al. (2022), Bayer et al. (2023)
Macroscopic material balances (4 species)	Shallow FFNN (tanh hidden nodes)	Levenberg-Marquardt; coupled	Yes	Optimize viable cell density	Nold et al. (2023)
Macroscopic material balances (4 species)	Shallow FFNN (tanh hidden nodes)	MATLAB *fminunc* function; coupled	Yes	Prediction of culture dynamics; Quality-By-Design	Narayanan et al. (2019)
Macroscopic material balances (6 species)	Gaussian Process regression	Maximum likelihood estimator; uncoupled	Yes	Prediction of culture dynamics across different products	Hutter et al. (2021)
Mechanistic kinetic models (12 species)	Deep FFNN with 2 hidden layers (softmax/sigmoid hidden nodes)	Python 3.7 Tensorflow/ gPROMS v.5.0.1; uncoupled	Yes	Prediction of culture dynamics and mAb glycosylation	Kotidis et al. (2021)

Macroscopic material balances (5 species)	Set of Shallow FFNN (tanh hidden nodes)	Levenberg-Marquardt; uncoupled	Yes	Software sensor of r-tPA production	Senger and Karim (2003)
Macroscopic material balances (4 species)	Principle Component Regression (PCR)	PCA + least squares regression; uncoupled	Yes	Prediction of culture dynamics	Okamura et al. (2022)
Macroscopic material balances (24 species)	Saturation and sigmoidal functions	Least squares regression; uncoupled		Automated assembly of dynamic model	Doyle et al. (2023)
CHO-K1 Genome-scale network (788 reactions; 686 species)	PCA of reaction rates of extracellular species	Linear programming; coupled	Yes	Hybrid FBA; Culture media design	Ramos et al. (2022)

Most previous hybrid modeling studies have combined material balance equations with shallow FFNNs or other nondeep machine learning techniques. In the field of neural networks, Deep neural networks have however been shown to have a general advantage over their shallow counterparts thanks to their ability to approximate more complex functions with a lower number of parameters and being less prone to overfitting (Delalleau and Bengio, 2011, Eldan and Shamir, 2016, Liang and Srikant, 2017, Mhaskar and Poggio, 2016). Training of deep structures also requires special care, with the ADAM method (Kingma, 2014) being commonly used due to its robustness and lower sensitivity to local optima. Along with the training approach, the use of stochastic regularization techniques has also been shown to be very effective at avoiding overfitting (Hinton et al., 2012, Srivastava et al., 2014, Koutsoukas et al., 2017).

Only very recently, hybrid modeling is incorporating deep neural networks and deep learning into its framework (Bangi and Kwon, 2020, Pinto et al., 2022, Bangi and Kwon, 2023). Pinto et al. (2022) investigated the use of ADAM and stochastic regularization in a hybrid modeling context concluding that the predictive power of deep hybrid models was significantly improved. None of these techniques have been applied to CHO processes (Table 8). In this study, we thus investigate deep learning techniques based on ADAM and stochastic regularization in a hybrid modeling context with application to a CHO-K1 fed-batch process. The

deep learning method is compared with the classical shallow method based on the LMM algorithm, indirect sensitivity equations and cross-validation.

5.2 Methods

5.2.1 CHO-K1 Experimental Dataset

Data from 24 fed-batch reactor experiments with a CHO-K1 cell line coding for a target glycoprotein were used to compare the hybrid modeling methodologies. Briefly, the cells were pre-cultured in shake-flasks (Corning, NY, USA) at 37°C in a proprietary chemically defined medium. The inoculum was transferred to 250 mL stirred microcarrier vessel (Ambr® 250 workstation, Sartorius, Göttingen, Germany) for antigen production. Stirring was kept at around 20 W/m^3. Dissolved oxygen was controlled at 30% of saturation by sparging pure oxygen. The pH was controlled at 7.0 with a 0.5M NaOH solution and CO2 sparging. The reactors were seeded at 3.0 Mcell/mL. They followed a batch/fed-batch phase for viable cells expansion. Once a threshold viable cell density was reached, the temperature was decreased to 33°C to induce antigen production. The antigen production phase was carried out in fed-batch mode with varying feeding compositions of amino acids, glucose, and pyruvate. The whole process lasted approximately 12 days. Samples were taken daily. Viable cell density and viability were assayed using a Vi-Cell cell counter (Beckman, Indianapolis, USA). Glucose, lactate, pyruvate, glutamine, ammonium, glycerol, and lactate dehydrogenase were assayed using a CedexBio-HT metabolite analyzer (Roche, Penzberg, Germany). The antigen quantification was performed off-line with an Octet HTX (Pall, NY, USA). The remaining metabolites and amino acids were assayed off-line by Nuclear Magnetic Resonance spectroscopy at Eurofins Spinnovation (Oss, The Netherlands). A total of 30 concentrations were measured at each time point (with few exceptions): viable cell count (Xv), glycoprotein (P), glucose (Glc), lactate (Lac), glutamine (Gln), glutamate (Glu), ammonium (NH4), pyruvate (Pyr), glycerol (Glyc), citrate (Cit), alanine (Ala), arginine (Arg), asparagine (Asn), aspartate (Asp), L-cystine (Lcystin), glycine (Gly), histidine (His), isoleucine (Ile), leucine (Leu), lysine (Lys), methionine (Met), phenylalanine (Phe), proline (Pro), serine (Ser), threonine (Thr), tryptophane (Trp), tyrosine (Tyr), valine (Val), acetate (Ac) and formate (For). The data was assumed to be corrupted by heterogenous gaussian noise. The measurement error standard deviations were assumed to be of 5% for P, 10% for Xv and 20% for remaining metabolites, based on equipment calibration data. The data reliability was pre-assessed by statistical analysis of metabolic fluxes in

the exponential growth and production phases. The spread of data was analyzed in a boxplot of metabolic fluxes. No outlying reactor experiments were identified. All the 24 reactor experiments were used for modeling thus none discarded due to reliability issues. More details regarding the experimental protocol and data pre-assessment are provided by Ramos et al. (2022)

5.2.2 CHO-K1 Synthetic Dataset

In addition to the experimental dataset, a synthetic dataset was created based on the metabolic model proposed by Robitaille et al. (2015). A synthetic dataset is useful in this context to better assess the ability of the hybrid modeling methods to describe the intrinsic process behavior irrespective of measurement noise. Simulations of this model were performed by varying two parameters, namely the pre-induction feeding rate and the post-induction feeding rate. A central composite design of experiments (CC-DoE) was applied to obtain 9 combinations of the two feed rates. This resulted in 9 fed-batch simulated experiments. The dynamic model has 21 intracellular species and 25 extracellular species. The intracellular species were hidden from the hybrid model development. The concentrations of extracellular species were recorded as time series for 240 hours with 24 hours sampling time and included the following variables: Xv, monoclonal antibody concentration (mAb), Ala, Arg, Asn, Asp, Cysteine (Cys), Glc, Gln, Glu, Pyr, Gly, His, Ile, Lac, Leu, Lys, Met, NH4, Phe, Pro, Ser, Thr, Tyr and Val. The recorded variables from the synthetic dataset were the same as in the experimental dataset, except that Pyr, Glyc, Cit and Ac are not considered in the Robitaille et al (2015) model. Moreover, the target products are different and Robitaille et al (2015) considers Cysteine instead of Cystine. Gaussian white noise with standard deviation of 10% of maximum concentration values was added to concentrations time points to mimic (heterogeneous) gaussian measurement error. This synthetic dataset is provided as supplementary material B.

5.2.3 CHO-K1 Hybrid Model

A standard hybrid model configuration was adopted in this study consisting of a multilayered FFNN connected in series with macroscopic material balance equations. This configuration is similar to previously published studies (Table 8) except for the depth of the FFNN and the training methods employed. The FFNN is dedicated to completely model the reaction kinetics. The dynamics of state variables are modeled by a system of ODEs based on macroscopic

material balance equations (First-Principles). Considering a perfectly mixed fed-batch bioreactor with multiple feed streams, the macroscopic material balance equations take the following state-space form (Equation 49):

Equation 49. State-space equations for the CHO-K1 hybrid model

$$\frac{d\boldsymbol{c}}{dt} = \boldsymbol{v}(\boldsymbol{c},\boldsymbol{w})X_v + \sum_k D_k \boldsymbol{c}_{k,in} - \boldsymbol{c}\sum_k D_k$$

$$\frac{dV}{dt} = V\sum_k D_k$$

$$D_k = \frac{F_k}{V}$$

with t the independent variable time, $\boldsymbol{c}$ the state vector with the concentrations of 30 species (Xv, P, Glc, Lac, Gln, Glu, Nh4, Pyr, Glyc, Cit, Ala, Arg, Asn, Asp, Lcystin, Gly, His, Ile, Leu, Lys, Met, Phe, Pro, Ser, Thr, Trp, Tyr, Val, Ac, For), $\boldsymbol{v}(\cdot)$ the specific reaction rates vector of the 30 species, $D = \sum_k D_k$ the reactor dilution rate (scalar), V the cultivation volume (scalar), F_k the feed rate of stream k (there are in total 5 feed streams) and $\boldsymbol{c}_{k,in}$ the vector of species concentrations in feed stream k. The specific reactions rates, $\boldsymbol{v}(\boldsymbol{c},\boldsymbol{w})$ lack mechanistic basis and were thus modeled by a deep FFNN with *nh* hidden layers (Equation 50):

Equation 50. General FFNN equation

$$\boldsymbol{H}^0 = \boldsymbol{c} \oslash \boldsymbol{c}_{max}$$

$$\boldsymbol{H}^i = \sigma\left(\boldsymbol{w}^i \cdot \boldsymbol{H}^{i-1} + \boldsymbol{b}^i\right), \qquad i = 1, \dots, nh$$

$$\boldsymbol{v} = \boldsymbol{w}^{nh+1} \cdot \boldsymbol{H}^{nh} + \boldsymbol{b}^{nh+1}$$

The input layer $i = 0$ with 30 nodes receives the information of normalized concentrations ($\boldsymbol{c}_{max}$ is the absolute maximum concentration of the 30 species (vector) and $\oslash$ the Hadamard division). Each hidden layer i computes a vector of outputs, $\boldsymbol{H}^i$, from a vector of inputs, $\boldsymbol{H}^{i-1}$, which are the outputs of the preceding layer. The transfer function of hidden nodes, $\sigma(\cdot)$, was either the hyperbolic tangent function, *tanh,* or the rectified linear unit, *ReLU*. The output layer computed the specific reaction rates vector of the 30 species. The parameters $\boldsymbol{w} = \{\boldsymbol{w}^1, \boldsymbol{w}^2, \dots, \boldsymbol{w}^{nh+1}\}$ are the nodes connection weights between layers and $\boldsymbol{b} = \{\boldsymbol{b}^1, \boldsymbol{b}^2, \dots, \boldsymbol{b}^{nh+1}\}$ the bias weights that need to be optimized data during the training process. The deep hybrid model Equation 49 and Equation 50 were integrated numerically using a Runge-Kutta 4th order ODE solver (in-house developed in MATLAB).

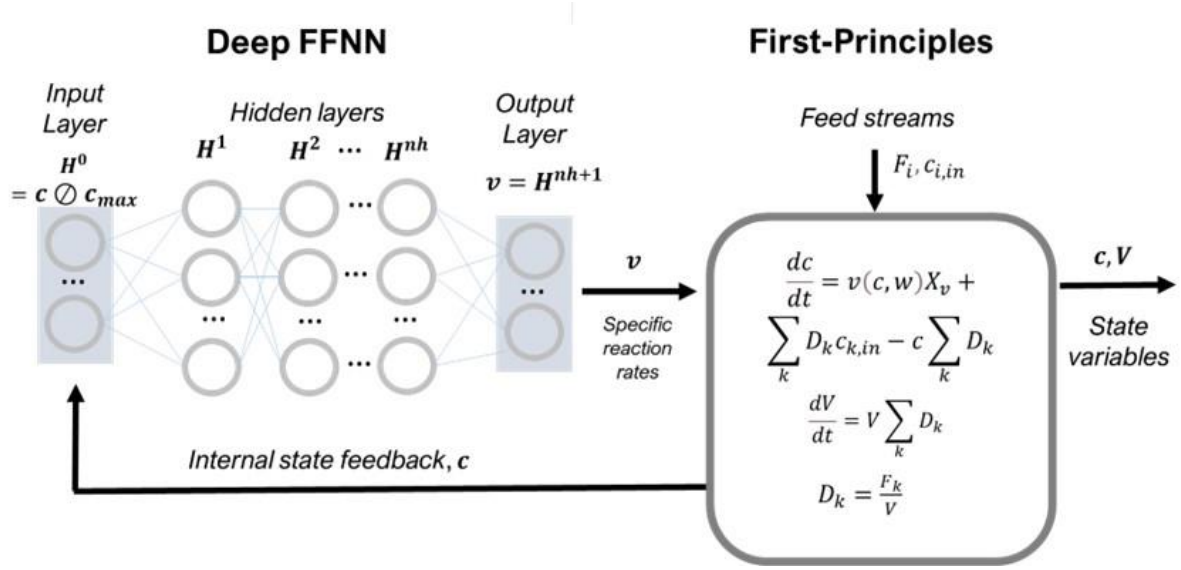

Figure 17. Hybrid model structure of a CHO-K1 fed-batch process.

5.2.3.1 Shallow Hybrid Modeling Method

This chapter compares shallow and deep hybrid modeling. The shallow structures are represented by Equation 49 and Equation 50 with FFNNs with a single hidden layer and with hyperbolic tangent activation function, *tanh*. Sigmoidal activation functions, and particularly *tanh*, are generally accepted as a default in shallow FFNNs. Many practical studies have corroborated the universal function approximation property derived by Cybenko (1987). This FFNN architecture has also been the preferred choice in a hybrid modeling context (e.g., Table 8). The training of shallow hybrid models is based on the LMM optimization with the indirect sensitivity equations (to compute gradients) and cross-validation (as early stop criteria). Briefly, the data were partitioned into a training/validation subset (for parameter estimation) and a testing subset (to assess the predictive power). Partitioning was performed batch wise with the amount of data allocated in each partition depending on the context (further details in section 5.3). The LMM algorithm (*fminunc* function in MATLAB) was adopted to optimize the network parameters, $\{\boldsymbol{w}, \boldsymbol{b}\}$, by unconstrained weighted least squares computed on the training data subset only. The inverse of measurement error variance was used as weighting factor in the weighted least squares minimization in order to effectively filter heterogeneous gaussian error Equation 51. The objective function gradients were computed by the indirect sensitivity equations following the method described by Oliveira (2004). Cross-validation was adopted as a stop criterion to avoid overfitting, i.e. the training is stopped when the validation error increases. A data augmentation strategy was used to automatically create the validation data subset from the training subset by adding gaussian noise to the concentrations

(Bejani and Ghatee, 2021). The standard deviation of the added noise was the same as the standard deviation of the measured concentration error. This strategy has proven to effectively avoid overfitting to the experimental noise and to produce good generalization models when the data information content is well distributed among the training and testing data subsets (Pinto et al. (2022)). For each shallow hybrid structure, the training was repeated 10 times with random weights initialization from the uniform distribution. Only the best result (lowest training/validation error) was kept.

5.2.3.2 Deep Hybrid Modeling Method

The shallow hybrid models were systematically compared with deep hybrid models. The deep hybrid models are represented by Equation 49 and Equation 50 with FFNNs with multiple hidden layers ($nh \geq 2$) and with rectified linear unit (*ReLU*) hidden nodes. The *tanh* was replaced by the *ReLU* because the latter is generally accepted as a default for several deep neural network architectures including deep FFNNs (Goodfellow et al., 2006). The *ReLU* function solved two main problems associated with the *tanh* function, namely signal saturation and the vanishing gradients problem that occurs during error backpropagation in networks with multiple hidden layers (Glorot and Yoshua, 2010). Instead of the LMM algorithm, deep hybrid models were trained with the ADAM algorithm (in-house implementation). The ADAM algorithm is generally accepted as an efficient method to train deep FFNNs (Kingma, 2014). The use of ADAM in a hybrid modeling context has been recently investigated by Pinto et al (2022). Briefly, the data were portioned in a training and in a testing subset as for shallow hybrid modeling. The ADAM was adopted to optimize the network parameter, $\{\boldsymbol{w}, \boldsymbol{b}\}$, also in a weighted least squares sense in order to effectively filter heterogeneous gaussian error Equation 51. The objective function gradients were computed by the semidirect sensitivity equations. The semidirect sensitivity equations method was shown to reduce the training CPU time in comparison to the indirect sensitivity equations method used in shallow hybrid modeling (Pinto et al (2022)). Stochastic regularization with minibatch size (0–1) and weights dropout probability (0–1) was applied to avoid overfitting in replacement of cross-validation normally applied in shallow hybrid modeling. The ADAM with stochastic regularization was run for a sufficiently large number of iterations with the final deep FFNN weights taken at the iteration with minimum training error. The training was performed only once because ADAM is less sensitive to weights initialization. This methodology has been thoroughly investigated by Pinto et al (2022).

5.2.3.3 Model Performance, Selection, and Implementation

The performances of shallow and deep hybrid models were assessed by the Weighted Mean Square Error (WMSE) computed as follows (Equation 51):

Equation 51. Objective function of the CHO-K1 model

$$WMSE = \frac{1}{T}\sum_{t=1}^{T}\frac{(c_t^* - c_t)^2}{\sigma_t^2}$$

with T the number of data examples, c_t^* the measured concentration at time t, c_t the model calculated concentration at time t and σ_t the standard deviation of measurement at time t. The WMSE was computed separately for the training and testing data subsets. In the case of the synthetic dataset, the test WMSE was computed using c_t^* with experimental noise (noisy test WMSE) and without noise (noise-free test WMSE).

Model selection was performed by a probabilistic method and by a resampling method. The probabilistic method consisted in the Akaike's Information Criterion (AIC) with second order bias correction (AICc). The second order correction is needed for small data samples (T<40), eventually converging to the AIC value for very larger samples (Banks and Joiner, 2017). It is computed on the training data subset as follows (Equation 52):

Equation 52. Akaike Information Criterion with second order bias correction

$$AICc = T\ ln(WMSE) + 2\ nw + \frac{2\ nw\ (\ nw\ +\ 1\ )}{T\ -\ nw\ -\ 1}$$

The AICc was adopted to discriminate parsimonious hybrid structures by taking into account the model complexity (i.e., the total number of network parameters, nw). The model with lowest AICc score was selected as the best model.

Model selection was also performed by a resampling technique. Ten different training and testing data partitions were created by random selection (from the uniform distribution) of reactor experiments allocated either for training or for testing. The training was repeated for every data partition resulting in 10 different models. The respective training and testing WMSE statistics were evaluated. The best model was selected to be the one with the lowest mean test WMSE.

The AICc and the resampling method often led to different model selection conclusions (further discussed section 5.3). It is generally accepted that resampling methods are preferred over probabilistic methods for statistical model selection (Tashman, 2000). Therefore, the resampling method, based on the lowest test WMSE, was taken as the final decision metric for the selection of hybrid models.

All the code of shallow and deep hybrid modeling was developed in-house and implemented in MATLAB on a computer with Intel(R) Core(TM) i5-8265U CPU @ 1.60 GHz 1.80 GHz, and 24 GB of RAM. CPU time of the different tests performed were computed as the difference between the result of the "*cputime*" MATLAB function at the start and end of a run.

5.3 Results

5.3.1 Shallow Hybrid Modeling of the CHO-K1 Synthetic Dataset

Shallow hybrid models with varying number of nodes in a single hidden layer with *tanh* activation function were investigated. At this stage, the synthetic dataset was adopted since it allows a better control of the information content distribution among the training and testing data subsets. The training partition was composed of 5 batches with 2400 training examples (the number of training examples was always higher than the number of FFNN weights). The testing partition was composed of 4 batches with 1920 testing examples. The training experiments were the center and square points of the CC-DoE, whereas the test experiments were the star points of the CC-DoE. The comparatively large testing data subset, generated at the extreme star points of the CC-DoE, represents a challenging extrapolating test for the trained hybrid models. Given the very clear testing rationale, the resampling repetitions were not applied in this case, which allowed to save some CPU time. The training and testing data subsets were always the same with models compared based on the AICc score and on the final test WMSE. The number of nodes of the hidden layer varied between 1 and 15 corresponding to a number of weights between 77 and 805. The training algorithm was the LMM with gradients computed by the indirect sensitivity method. For each structure, the training was repeated 10 times with different weights initialization (classical method). The overall results are shown in Table 9. These results confirm that the number of nodes in the hidden layers has a significant effect on the model performance. The AICc score and the test WMSE did not converge to a common conclusion (discussed below). The shallow structure with lowest AICc had 5 hidden nodes only, which did not correspond to the lowest test error. The shallow structure with highest predictive power had 12 hidden nodes with the lowest noisy and noise-free test WMSE (2.04 and 2.06, respectively). The noisy test WMSE was 32.5% higher than the train WMSE denoting some degree of overfitting of the training data. The AICc criterion miss selected the model with the highest predictive power in this case.

Table 9. Shallow hybrid modeling results on the CHO-K1 synthetic dataset. Hybrid models had a FFNN with a single hidden layer with hyperbolic tangent activation function and a number of nodes between 1 and 15. The

training algorithm was the Levenberg-Marquardt with gradients computed by the indirect sensitivity equations with 1000 iterations and cross-validation as stop criterion. Training was repeated 10 times for each structure with random weights initialization from the uniform distribution between -0.01 and 0.01 and only the best result was kept. The WMSE-train was computed on the training dataset with 10% gaussian noise in concentrations. WMSE-test (noisy) was computed on the test dataset with 10% gaussian noise in concentrations. WMSE-test (noise free) was computed on the test dataset without noise in the concentrations. The AICc was computed on the same dataset as WMSE-train.

Number of hidden nodes	WMSE - train	WMSE-test (noisy)	WMSE-test (noise free)	AICc	CPU time (hh:mm:ss)	Number of weights
1	6.07	7.46	8.16	4890	00:13:20	77
2	2.17	3.82	4.32	2310	00:25:31	129
3	1.81	3.25	3.64	1950	00:30:04	181
4	1.76	2.79	3.15	2000	00:26:44	233
5	**1.28**	**4.57**	**4.31**	**1290**	**00:23:06**	**285**
6	1.52	2.31	2.76	1890	00:27:34	337
7	1.55	2.10	2.18	2070	00:24:58	389
8	1.66	3.09	3.45	2400	00:30:18	441
9	1.73	2.71	2.79	2500	00:26:40	493
10	1.60	2.47	2.63	2450	00:32:20	545
11	1.70	2.73	3.12	2930	00:28:15	597
12	**1.54**	**2.04**	**2.06**	**2850**	**00:24:52**	**649**
13	1.64	2.70	2.84	3210	00:32:30	701
14	1.73	6.33	7.14	3550	00:18:15	753
15	1.54	2.65	2.86	3460	00:22:18	805

5.3.2 Deep Hybrid Modeling of the CHO-K1 Synthetic Dataset

Deep hybrid modeling with FFNNs with 2 or 3 hidden layers was investigated on the same synthetic dataset. Models with more than 3 hidden layers did not produce further improvements (results not shown). The activation function in the hidden layer was the *ReLU* in all cases. The training algorithm was the ADAM with standard hyperparameters (Kingma, 2014). Stochastic regularization with optimal minibatch size of 0.8 and weights dropout of 0.2 was adopted, based on a previous study by Pinto et al. (2022). Stochastic regularization coupled with ADAM was shown to be very robust to weights initialization (Pinto et al., 2022) thus the training was carried out only once with a single random weights initialization (between -0.01 and 0.01). The overall results are shown in Table 10. As expected, the complexity of the FFNN has a significant effect on the model performance. The number of weights varied between 315–1905, always lower than the number of training examples (2400). The hybrid structure 10×10×10 with 765 weights clearly stands out as the best performing structure. The obtained training and testing errors are comparable denoting a successful training without overfitting. Moreover, the noise free test error is clearly below the noisy test error, showing that this model was able to filter noise in the test partition. The AICc of the 10×10×10 structure was also the lowest among the deep hybrid structures investigated. The AICc and the test WMSE pointed to the same conclusion in this case.

Table 10. Deep hybrid modeling results on the synthetic CHO-K1 dataset. Hybrid models had a FFNN with 2 or 3 hidden layers with *ReLU* activation function. The training algorithm was the ADAM algorithm run for 1000 iterations with hyperparameters α=0.001, β1=0.9 β2=0.999 and η=1e^(-7). Gradients were computed by the semidirect sensitivity equations. Stochastic regularization was applied with weights dropout of 0.2 and minibatch size of 0.8. The training was repeated only once with random weights initialization from the uniform distribution between -0.01 and 0.01. The WMSE-train was computed on the training dataset with 10% gaussian noise in concentrations. WMSE-test (noisy) was computed on the test dataset with 10% gaussian noise in concentrations. WMSE-test (noise free) was computed on the test dataset without noise in the concentrations. The AICc was computed on the same dataset as WMSE-train.

Number of hidden nodes	WMSE train	WMSE test (noisy)	WMSE test (noise free)	AICc	CPU time (hh:mm:ss)	Number of weights
[5 5]	1.85	2.47	2.63	2330	00:14:20	315
[7 7]	1.48	2.00	1.94	2090	00:13:30	445
[10 10]	1.34	1.84	1.56	2510	00:17:15	655

[5 5 5]	2.00	4.43	4.35	2610	00:19:43	345
[7 7 7]	1.50	2.13	2.35	2300	00:15:18	501
[10 10 10]	**0.982**	**1.05**	**0.54**	**1800**	**00:19:42**	**765**
[15 15]	1.33	1.72	1.62	3970	00:17:32	1045
[20 20]	0.922	1.27	1.01	4250	00:22:51	1485
[20 20 20]	0.972	1.27	0.98	6860	00:24:47	1905

Comparing the shallow hybrid model with 12 hidden nodes (Table 9) with the deep hybrid model with 3 hidden layers (10×10×10) (Table 10) shows that the latter has significantly better training and testing metrics. The 3 hidden layers did not correspond to a large increase in the number of weights (only 17.9%). However, the training error decreased 36.2% and more importantly the noise free test error decreased 73.8%. Both the AICc score and the test WMSE point to the hybrid deep structure (10×10×10) as being the best model. As for the CPU time, albeit the higher complexity of the deep model (with 17.9% more parameters), the CPU time was reduced by 20.8%. This is mainly explained by the fact that ADAM with stochastic regularization is practically insensitive to weights initialization requiring a single training event compared to the 10 training repetitions in the case of LMM with cross-validation.

Figure 18 shows the prediction of the dynamics in a test experiment by the best shallow and best deep hybrid models. This example shows qualitatively that the deep hybrid model succeeded to predict very faithfully the dynamics of each variable individually (the predicted time profiles of process variables are always within the error bars). Conversely, the shallow hybrid structure shows systematic deviations in different process phases for different variables. As examples, mAb, Ala, Cys, Gly, Asn, Glu and Thr show systematic deviations in relation to the true profiles.

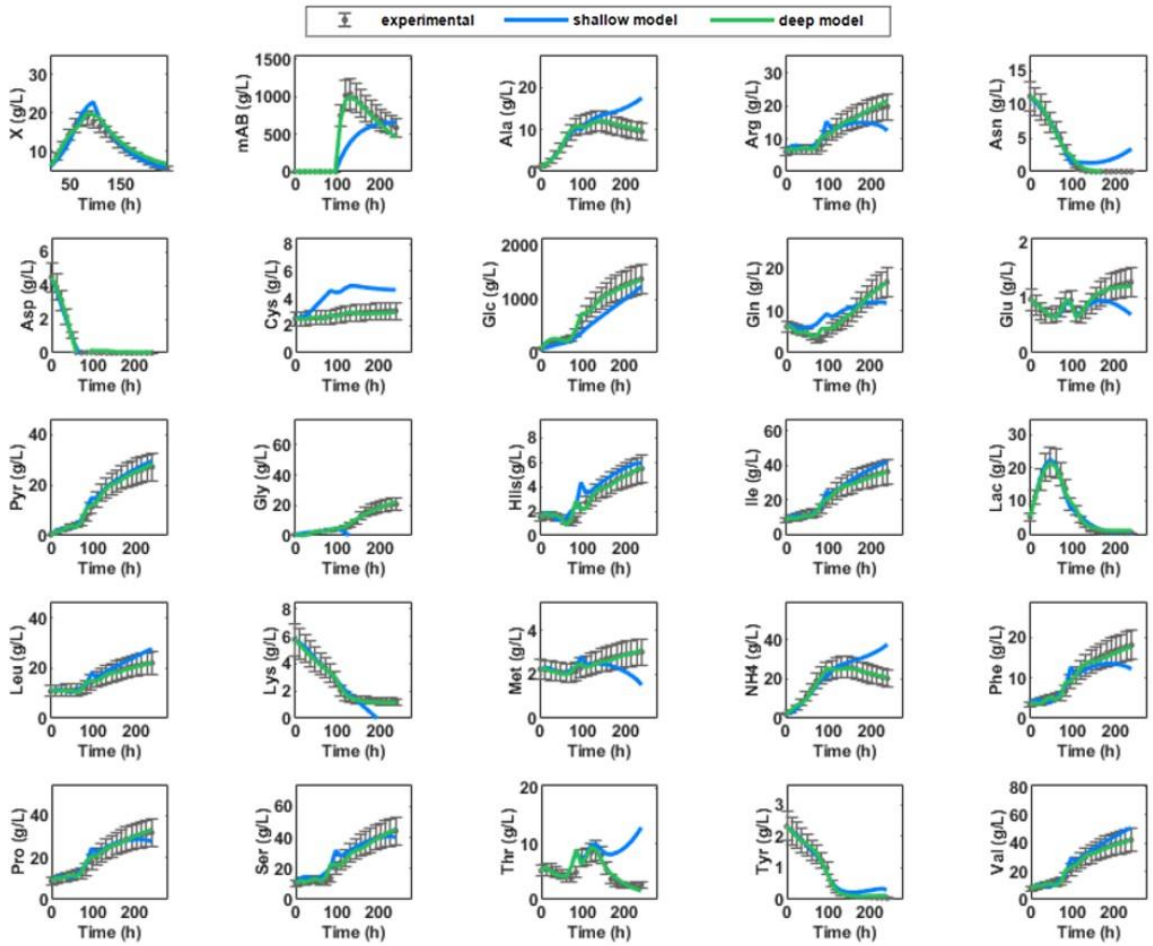

Figure 18. Dynamic simulation of the best shallow (12 hidden modes + *tanh*) and best deep (10x10x10 + *ReLU*) hybrid models for a test reactor experiment of the CHO-K1 synthetic dataset. Circles are simulated data points and error bars are standard deviations. Green line is the best deep hybrid model structure (10×10×10) (Table 10); Blue line is the best shallow structure with 12 hidden nodes (Table 9).

5.3.2.1 Comparison Between Training Methods

In order to better understand if the differences in the models performances are due to the training method or to the depth of the FFNNs, the shallow hybrid structures of Table 9 were also trained with the deep learning method (ADAM + semidirect sensitivity + stochastic regularization) and the deep structures of Table 10 were also trained with the classical method (LMM + Indirect sensitivity + cross-validation). The results are shown in Figure 19. Figure 19A shows that the final training error is comparable for both methodologies in the case of shallow hybrid models. The testing error tends to be slightly lower and more stable for shallow hybrid models trained with ADAM. The LMM delivers in some cases equally performing models, but it is more unstable. For deep hybrid models with 2 (Figure 19C, Figure 19D) and 3 (Figure 19E, Figure 19F) hidden layers, the differences between both methods are more

substantial. For deep structures, as the model size increases the training and testing errors of the ADAM method are significantly lower than those of the LMM method. For large models (number of weights approaching 2000), the difference between ADAM and LMM final training and/or testing errors is as high as 100%. Contrary to ADAM, the final training error delivered by LMM tends to increase with the number of weights suggesting that this approach is unable to exploit the descriptive power of deep FFNNs. However, for small FFNN structures the LMM performs equally or better than the ADAM method.

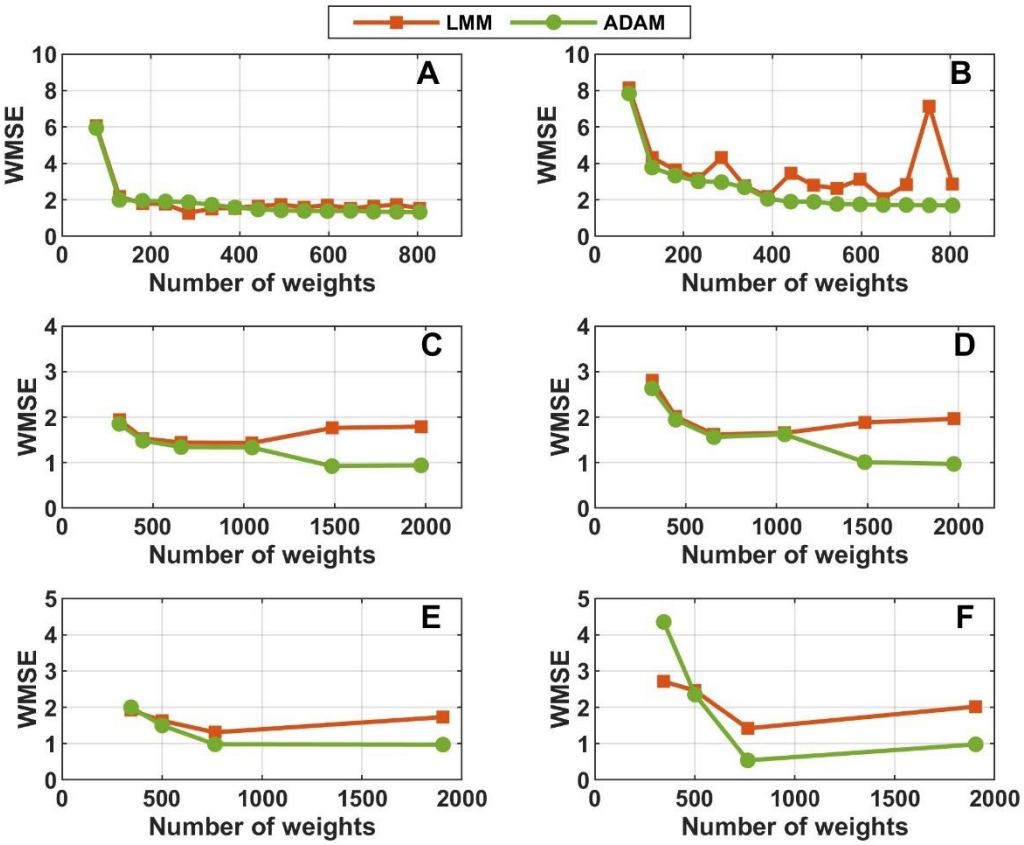

Figure 19. Hybrid model final training and testing errors as function of the FFNN depth (number of hidden layer) and size (number of weights). Orange line and orange squares – hybrid models trained with LMM + indirect sensitivity equations + cross-validation. Green line and green circles – hybrid models trained with ADAM + semidirect sensitivity equations + stochastic regularization. **A)** Training WMSE of shallow hybrid models (Table 9). **B)** Testing WMSE of shallow hybrid models (Table 9). **C)** Training WMSE of hybrid models with 2 hidden layers (Table 10). **D)** Testing WMSE of hybrid models with 2 hidden layers (Table 10). **E)** Training WMSE of hybrid models with 3 hidden layers (Table 10). **F)** Testing WMSE of hybrid models with 3 hidden layers (Table 10).

5.3.3 Hybrid Deep Modeling of the CHO-K1 Fed-Batch Process

The hybrid modeling framework was applied to the 24 fed-batch experiments collected in a process development campaign to produce a therapeutic glycoprotein. Deep hybrid structures with 2 or 3 hidden layers with nodes between 3 and 30 were investigated. For comparability, single hidden layer hybrid models with 1 to 18 nodes were also investigated. Given the results of the previous section, only the deep learning method based on ADAM, semidirect sensitivity equations and stochastic regularization was adopted. The training hyperparameters were kept the same as in the synthetic dataset study. The training partition was composed in this case of 20 experiments with 7953 training examples (83% of data). The test-

ing partition was composed of 4 batches with 1593 testing examples (17% of data). The training was repeated 10 times for each hybrid model structure with random permutations of test/train experiments to avoid data selection bias, with the results analyzed statistically (resampling method). The 10 train/test permutations were kept the same in all tests performed to ensure comparability. The overall results are shown in Table 11. Structures with less than 8 hidden nodes did not have sufficient complexity to describe the process, showing a very high and unstable training error. The hybrid deep structure (25×25×25) with 2855 parameters showed the lowest test error of 1.88±0.44, although 39.3% higher than the training error (1.35±0.21). The best shallow structure with 17 hidden nodes had 16.3% higher training error and more importantly 30.8% higher test error compared to the best deep structure. As in the previous sections, increasing the depth of the FFNN seems to be advantageous in terms of predictive power. The lowest AICc was obtained with the structure (25×25×25) which also had the lowest test error.

Table 11. Hybrid modeling results on the experimental CHO-K1 dataset with 24 independent fed-batch experiments and 31 state variables. The activation function in the hidden layers was the *ReLU* in all cases. Hybrid models were trained with ADAM ($\alpha = 0.001$, $\beta 1 = 0.9$, $\beta 2 = 0.999$ and $\eta = 1e^{-7}$), semidirect sensitivity equations and stochastic regularization (minibatch size = 0.8 and weights dropout = 0.2). For each structure, the training was repeated 10 times with random train/test experiment permutations. Error metrics (WMSE-train, WMSE-test and AICc) are displayed as the mean ± SD of the 10 repetitions.

Number of hidden nodes	WMSE-train	WMSE-test	AICc	CPU time (hh:mm:ss)	Number of weights
7	Unstable	Unstable	Unstable	Unstable	457
8	25.9±0.74	33.6±1.14	70000±220	01:32:00	518
9	7.39±0.65	9.18±0.89	24000±150	01:37:00	579
10	3.54±0.40	4.12±0.75	9075 ± 120	01:40:00	640
11	3.11±0.36	4.09±0.41	6980±80	02:05:00	701
12	2.61±0.28	3.84±0.62	4650±60	01:52:00	762
13	1.74±0.29	2.88±0.62	3920±70	02:01:00	823
14	1.68±0.27	2.74±0.55	3880±75	02:10:00	884
15	1.60±0.28	2.66±0.54	3790±60	02:15:00	945
16	1.58±0.28	2.51±0.50	3775±80	02:17:00	1006

17	1.57±0.27	2.46±0.42	3800±70	02:21:00	1067
18	1.58±0.27	2.47±0.45	4025±70	02:18:00	1128
[5 5]	Unstable	Unstable	Unstable	Unstable	365
[7 7]	17.4±0.61	26.9±0.76	50000±200	01:22:00	513
[10 10]	1.57±0.25	2.50±0.77	3950±75	01:38:00	750
[5 5 5]	Unstable	Unstable	Unstable	Unstable	395
[7 7 7]	4.61±0.31	5.61±0.69	14010±100	01:16:00	569
[10 10 10]	1.41±0.21	2.17±0.55	3750±61	02:13:00	860
[15 15]	1.45±0.22	2.33±0.45	3890±70	02:21:00	1185
[20 20]	1.39±0.25	2.10±0.51	3730±80	02:33:00	1670
[25 25]	1.38±0.21	2.03±0.49	3725±60	02:49:00	2205
[30 30]	1.34±0.23	1.98±0.43	3630±70	02:59:00	2790
[20 20 20]	1.37±0.22	2.00±0.48	3680±70	02:41:00	2090
[25 25 25]	**1.35±0.21**	**1.88±0.44**	**3625±60**	**03:05:30**	**2855**
[30 30 30]	1.34±0.23	1.95±0.42	3715±80	03:43:00	3720

5.3.4 Predictive Power Analysis of the Hybrid Deep Structure (25×25×25)

The best deep hybrid structure (25×25×25) was analyzed in more detail. Figure 20A shows the training and test errors obtained for the 10 train/test permutations. The partitioning of data for training and testing has indeed a significant effect on the modeling error metrics. Partition 1 produced a low training error but also the highest test error. Partition 2 produced the best results with both low training and testing errors, and closely matching each other. These results show that the process information content is not equally distributed among the 10 randomly selected train/partitions. This problem can be mitigated with more data added to both the train and test partition in the future. Figure 20B further details model predictions of all concentrations over the respective experimental values for partition 8, which had the closest train and test error to the respective mean values. The slope of the linear regression as well as the Pearson correlation coefficient (r^2) of train and test data are similar. This shows

that despite the slightly larger WMSE for the test partition, there is no significant bias when compared to the train partition data subset.

A)

Partition	WMSE train	WMSE test
1	1.10	2.88
2	1.25	1.31
3	1.33	2.24
4	1.04	1.82
5	1.25	1.67
6	1.60	2.29
7	1.55	1.55
8	1.44	1.68
9	1.70	1.71
10	1.23	1.64
Mean	**1.35**	**1.88**
Standard Deviation	**0.21**	**0.44**

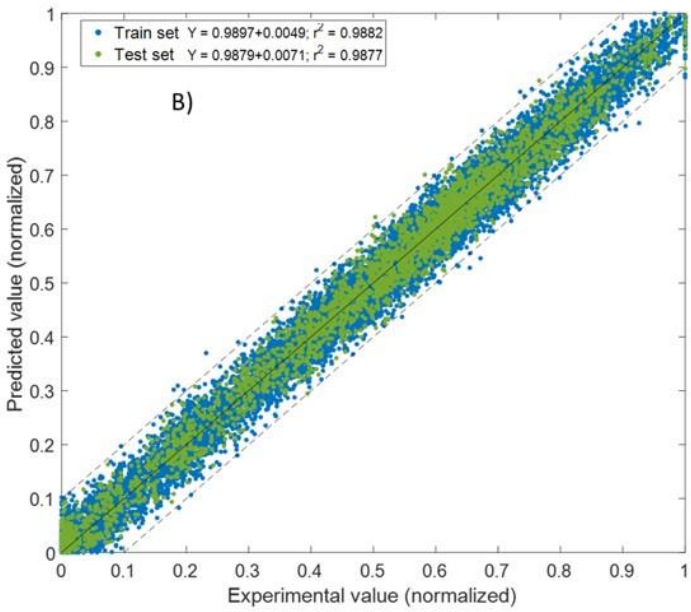

Figure 20. Training results for the best hybrid model structure (25×25×25) with 2855 weights. **A** – Final training and testing error for 10 randomly selected train (20)/test (4) permutations of experiments. **B** - Predicted over measured concentrations of all biochemical species for training/test partition 8 (highlighted in A). Blue circles are training data. Green circles are test data. Full line is the linear regression. Dashed lines are the upper and lower intervals corresponding to one standard deviation. The r^2 is the Pearson correlation coefficient.

The predicted time profiles were analyzed qualitatively for each variable individually. Figure 21 shows the dynamic profiles of the 30 concentrations individually for a selected test experiment (experiment 8) predicted by the best shallow model with 17 hidden nodes and the best deep model (25×25×25) trained on partition 8. The deep hybrid model follows very closely the measured data. Particularly, viable cells (Xv) and product (P) were accurately predicted. The predictions of metabolites are within the experimental error bars or very close. On the contrary, predictions of the best shallow hybrid model show a tendency to deviate outside of experimental error bounds, especially as the cultivation progresses in time. Figure 22 shows the predicted time profiles for several test experiments for a subset of process variables. It shows that viable cell count, glycoprotein titer, glucose and glutamine concentrations are always predicted within the error bars. Moreover, the switch between lactate production and lactate consumption as well as from ammonium production and ammonium consumption were correctly described by the model.

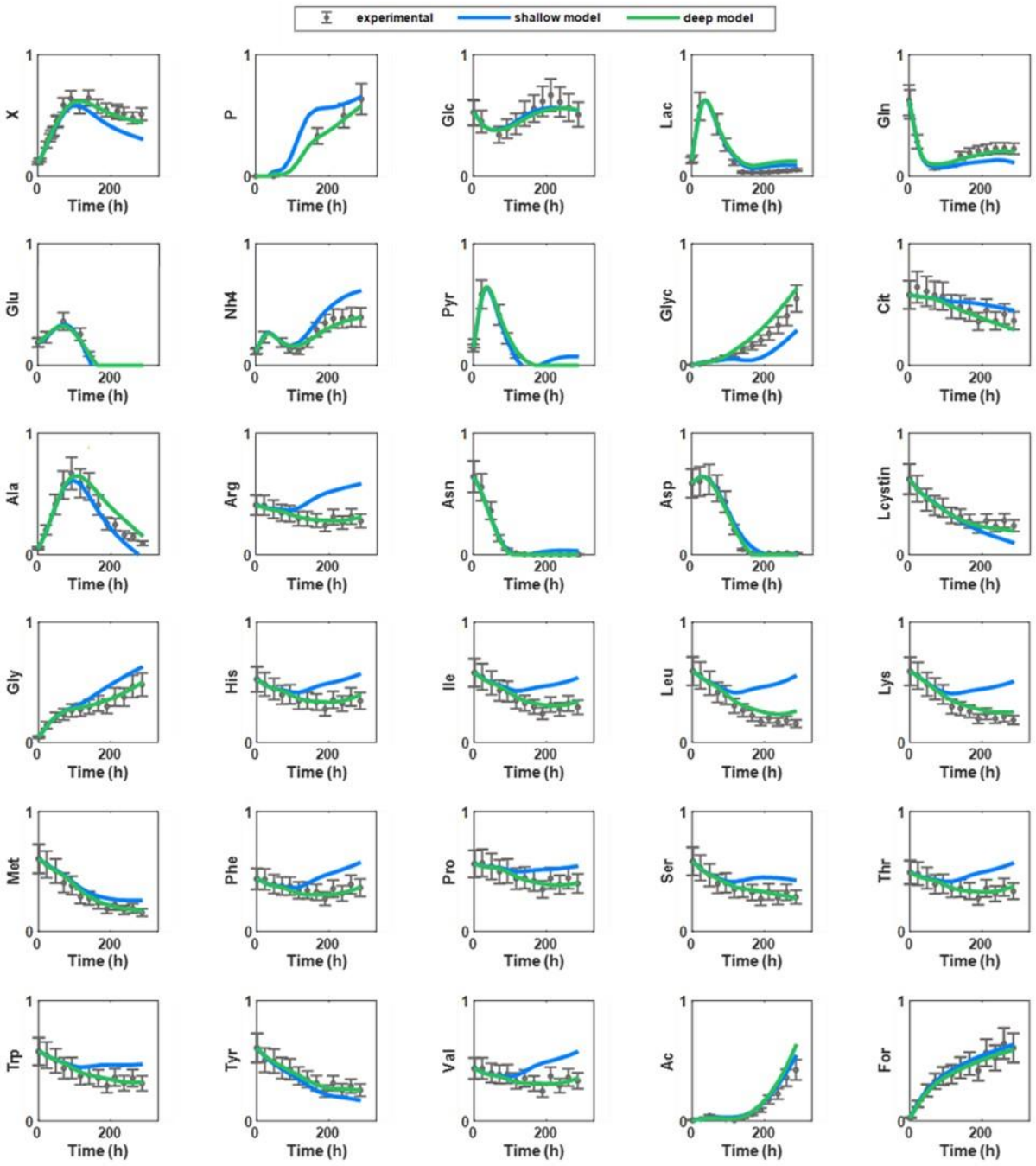

Figure 21. Dynamic simulation of best shallow (17) and best deep hybrid (25×25×25) models for a test experiment of the CHO-K1 experimental dataset. Circles are experimental data points and error bars are measurement standard deviation. Green line is the best deep hybrid model structure 25×25×25; Blue line is the best shallow hybrid structure with 17 hidden nodes.

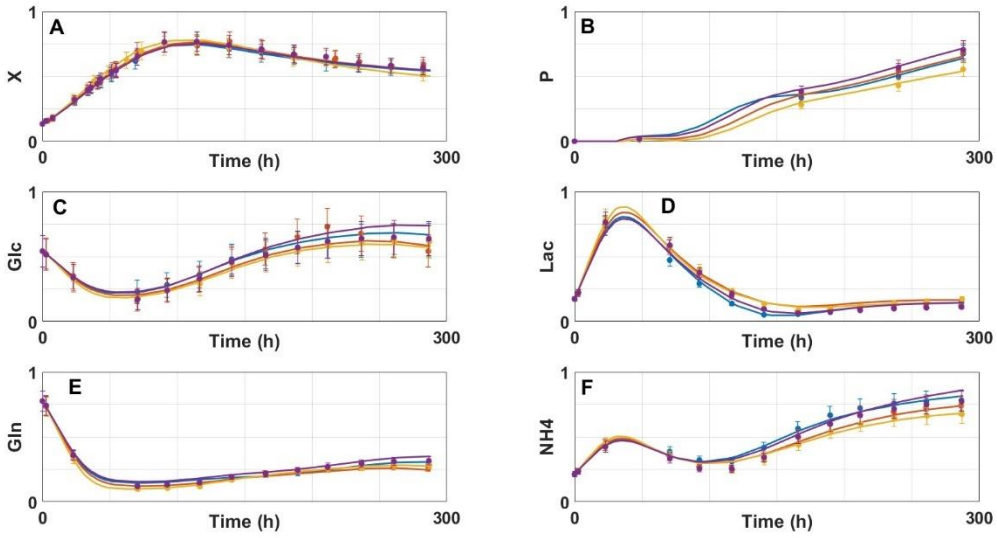

Figure 22. Dynamic simulation of best deep hybrid model (25×25×25) for multiple test experiments of the CHO-K1 experimental dataset. Circles are experimental data points and error bars are measurement standard deviation. Full lines are model predictions. The color code (symbol + full line) refers to different test experiments of partition 8. Blue, orange, yellow and purple colors represent test experiments 1, 4, 5 and 8 respectively. **A** – viable cell count. **B** – glycoprotein titer. **C** – glucose concentration. **D** – lactate concentration. **E** – glutamine concentration. **F** – ammonium concentration.

5.4 Discussion

Hybrid modeling combining First-Principles with neural networks is a well-established methodology in process systems engineering since the early 90's (e.g. von Stosch et al., 2014, Agharafeie et al., 2023). Only very recently hybrid modeling is incorporating deep neural networks and deep learning into its framework (Bangi and Kwon, 2020, Pinto et al., 2022, Bangi and Kwon, 2023). Most hybrid modeling studies of CHO cells followed the shallow approach. The primary goal of this chapter was to investigate if hybrid deep modeling is advantageous over shallow hybrid modeling in a CHO-K1 process development context.

5.4.1 Is Deep Hybrid Modeling Advantageous?

In the case of the synthetic dataset the best shallow model had (12) hidden nodes (Table 9) whereas the best deep structure had 3 hidden layers (10×10×10) (Table 10). The deep model complexity, as measured by the number of weights, increased only 17.9% in relation to the shallow model. The deep structure achieved a reduction of 36.2% in the training error (WMSE-train), 48.5% in the test error (WMSE-test noisy) and 73.8% in the noise free test error

(WMSE-test noise free). All error metrics were significantly improved with emphasis on the noise-free test error, which clearly shows that the deep structure captured more faithfully the intrinsic process dynamics. The CPU time was also reduced by 20.8%. It is noteworthy to mention that the Robitaille et al. (2015) model used to generate the synthetic dataset included the intracellular dynamics of 21 molecular species. The cells accumulated different amounts of intracellular species depending on the reactor feeding conditions eventually triggering different regulatory mechanisms. The deep FFNN is of static nature thus a structural bias could be anticipated due to the mismatch between the dynamic nature of the true process and the structure of the hybrid model. This was however successfully mitigated as reflected in the extremely low noise free test error of extracellular concentrations (Table 10 and Figure 18).

In the case of the experimental dataset the best shallow model had 17 hidden nodes whereas the best deep structure had 3 hidden layers (25×25×25) (Table 11). The model complexity (number of weights) increased in this case quite substantially by 167.6%. The deep structure achieved a reduction of 14.0% in the training error (WMSE-train) and 23.6% in the test error (WMSE-test) on average. In this case it is impossible to evaluate the noise-free test error reduction. Although the magnitude of the improvement is lower than in the synthetic dataset, it is statistically significant. Moreover, the improvement in the test error is on average higher than in the training error. The training CPU time increased in this case by 31.6%. This increase is explained by the higher model complexity (more 167.6% weights). It becomes clear that CPU time increase does not scale linearly with model complexity (number of weights). This is related with the computation of gradients by the semidirect sensitivity equations (Pinto et al., 2022). In this approach, the sensitivity of state variables in relation to network outputs are independent of the size of the network.

The results obtained for both the synthetic and experimental datasets indicate a clear advantage of deep hybrid models over shallow hybrid models in terms of predictive power. In both cases the test error reduction is significant and always higher than the training error reduction. This suggests that hybrid deep structures capture more faithfully the intrinsic nonlinear dynamics of the true process than the shallow counterpart when exposed to the same training dataset. This eventually translates into more accurate predictions of novel process conditions. This advantage is generally accepted for standalone FFNNs (Goodfellow et al. (2006) and is likely to generalize for hybrid models incorporating deep FFNNs. The only downside to the deep model in this study is the training CPU time increase. Pinto et al. (2022) reported a decrease in prediction error of 18.4% in a *Pichia pastoris* pilot process using the

same training scheme, which is close to the one reported here. In that study, the shallow and deep structures had the same number of weights, and as such the CPU time was also decreased by 43.4%. The CPU cost comparison seems to be case dependent and mainly related with the size of the shallow and deep FFNN embodied in the hybrid model.

5.4.2 What is the Best Training Method?

Two different training methodologies were compared in this study: the classical method and the deep learning method. The classical method is based on the LMM algorithm coupled with indirect sensitivity equations and cross-validation. This method is normally used to train shallow hybrid models (Table 8). The LMM is prone to be trapped in local optima. For this reason, the training must be repeated several times (in our case 10 times) with different parameter initializations for each structure investigated. The deep learning method is based on ADAM, semidirect sensitivity equations and stochastic regularization. ADAM is an improvement of the stochastic gradient descent algorithms with adaptive learning rate. The method estimates the learning rate during the training, based on the first and second moments of the gradients (Kingma, 2014). Only very recently ADAM was applied to train hybrid models (Pinto et al., 2022). A key conclusion was that ADAM is less prone to be trapped in local optima and is practically insensitive to weights initialization. For this reason, the ADAM training was repeated only once for each of the structures investigated, which in theory reduces the CPU time for FFNNs of comparable sizes. Based on the results of Figure 19 with the synthetic dataset, the ADAM method outperforms the classical method based on LMM both in terms of the training and test error especially for deep and large FFNNs. The differences are less marked for shallow and small FFNNs.

5.4.3 What is the Optimal Network Complexity?

Several methods have been proposed to determine the optimal neural network size (Teoh et al., 2006, Mohanan et al., 2022, Lawrence et al., 1997, Lawrence et al., 1996) but there is no consensus on a general methodology. Here, the number of hidden layers and number of nodes in hidden layers were chosen heuristically starting with a single hidden layer with a number of nodes equal to approximately half the number of inputs and then increasing until the optimal size is found. This procedure is replicated with an increasing number of hidden layers. Adding nodes and layers obviously carries a higher number of weights and higher complexity. Thus, choosing the best structure must balance the decrease in error with the

increase in model complexity. It is noteworthy to mention that the AICc criterion, which is evaluated on the training dataset only, often fails to discriminate the hybrid structures with the lowest test error. This is an important point because the final hybrid model is expected to faithfully predict unseen process conditions. Unseen process conditions mean that the test data is not yet available. Mei and Smith (2021) have compared probabilistic methods (the AIC and the Bayesian Information Criteria (BIC)) with a resampling method based on blocked cross-validation for selection of shallow FFNNs trained on meteorological data. They concluded that these approaches do not converge to the same conclusions, with the AIC and BIC generally selecting simpler models than the resampling technique. The results in this study show that the AICc and the resampling methods pointed roughly to the same conclusions in the case of hybrid models trained with ADAM (Table 10 and Table 11). This means that the lowest AICc score, calculated solely on the training dataset, coincided with the lowest test error statistics produced by the resampling method. Both methods selected the hybrid deep structure (25×25×25) in the case of the experimental dataset (Table 11) and the hybrid deep structure (10×10×10) for the case of the synthetic dataset (Table 10). The AICc failed however to discriminate the shallow hybrid model with the lowest test error in the case of the synthetic dataset and the LMM training method (Table 9). It clearly selected a much simpler model in line with the results by Mei and Smith (2021). It is generally accepted that the performance of statistical models should be assessed using resampling methods rather than probabilistic methods (Tashman, 2000). It is thus advisable to apply resampling methods also in the context of hybrid modeling despite the higher CPU cost. In both cases (synthetic and experimental datasets) the optimal depth was 3 hidden layers.

5.5 Conclusions

This chapter compares for the first time deep and shallow hybrid modeling of a CHO-K1 fed-batch process in a process development campaign. Data of a CHO-K1 cell line expressing a target glycoprotein comprising 24 independent fed-batch experiments with 30 measured state variables were used to compare both methodologies. The results point to a systematic generalization improvement of deep hybrid models with FFNNs with 3 hidden layers over shallow hybrid models. The overall improvement was 14.0% in the training error and 23.6% in the testing error. The CPU time to train the deep hybrid model increased by 31.6% and is mainly related to the higher FFNN complexity. It is today generally accepted that deep neural networks have a general advantage over their shallow counterparts in terms of descriptive

power and generalization capacity. This study points to a similar conclusion in a hybrid modeling context. Particularly, deep hybrid models tend to generalize better than shallow hybrid models provided that efficient deep learning algorithms (such as ADAM with stochastic regularization) are adapted to the hybrid model framework. This study focused on FFNN hybrid structures. The combination of first Principles equations with more complex deep neural network architectures, such as convolution neural networks (CNN) and long short-term memory (LSTM) networks, are future research directions in the hybrid modeling field. Shallow hybrid modeling is currently a method of choice in the digitalization of biopharma processes. We expect deep hybrid modeling to further accelerate the deployment of high-fidelity digital twins in the biopharma sector in the near future.

6

A GENERAL HYBRID MODELING FRAMEWORK FOR SYSTEMS BIOLOGY APPLICATIONS

This chapter is based on the publications: Pinto, J., Ramos, J. R., Costa, R. S., & Oliveira, R. (2023). A General Hybrid Modeling Framework for Systems Biology Applications: Combining Mechanistic Knowledge with Deep Neural Networks under the SBML Standard. *AI*, *4*(1), 303-318. and Pinto, J., Costa, R. S., Alexandre, L., Ramos, J., & Oliveira, R. (2023). SBML2HYB: a Python interface for SBML compatible hybrid modeling. *Bioinformatics*, *39*(1), btad044.

6.1 Introduction

Hybrid modeling methods combining mechanistic knowledge with machine learning (ML) in a common workflow have found wide application in process systems engineering since the early 1990s (e.g., review by von Stosch et al., 2014). Psichogios and Ungar (1992) described one of the first applications of hybrid models to bioprocess engineering. The proposed hybrid model consisted of dynamic material balance equations of biochemical species (system of ordinary differential equations (ODEs)) connected with a shallow feed-forward neural network in a common mathematical structure. Sensitivity equations were derived enabling the training of the neural network by error backpropagation on indirect training examples (e.g., measured target variables not coincident with the neural network output variables). Thompson and Kramer (1994) framed this problem as hybrid semiparametric modeling, as such models merge parametric functions (stemming from knowledge) with nonparametric functions (stemming from data) in the same mathematical structure. Schubert et al. (1994) presented the first industrial application of hybrid modeling (material balance equations combined with neural networks) to a Baker's yeast process. Since the early 1990s, hybrid model

structure definition, parameter identification and model-based process control have been extensively covered (e.g., Teixeira et al., 2006; Teixeira et al., 2007; von Stosch et al., 2011; Pinto et al., 2019; Rajulapati et al., 2022; Glassey and von Stosch, 2018). Hybrid models were applied to a wide array of microbial, animal cells, mixed microbial and enzyme processes in different industries, such as wastewater treatment, clean energy, biopolymers, and biopharmaceutical manufacturing (Agharafeie et al., 2023). The potential advantages of hybrid modeling may be summarized as a more rational usage of prior knowledge (mechanistic, heuristic, and empirical) eventually translating into more accurate, transparent, and robust process models (von Stosch et al., 2011; Glassey and von Stosch, 2018).

With a significant lag, hybrid modeling is currently receiving a lot of attention in the systems biology scientific community. ML has been applied for the prediction of the function of genes (Le at al., 2020) and proteins (Le, 2022) and is gaining popularity in all fields of systems biology (Greener at al., 2022). Cuperlovic-Culf et al. (2023) highlighted the difficulty of gathering high-quality *in vivo* data to validate detailed metabolic models, and the opportunity to alternatively apply ML and hybrid mechanistic/ML methods. Antonakoudis et al. (2020) recently reviewed the efforts to integrate GEnome-scale Models (GEMs) with supervised and unsupervised ML. Kim et al. (2021) reviewed ML applications in the construction and simulation of GEMs, and ML applications in use of GEM-derived information. The integration of mechanistic models and ML may be realized through a hybrid pipeline of activities, where both modeling frameworks participate to solve particular sub-tasks. Alternatively, mechanistic and ML models may be "fused" in a common semiparametric mathematical structure. Following the latter approach, hybrid metabolic flux analysis, combining metabolic networks and principal component analysis (PCA) in semiparametric linear models, has been studied by Carinhas et al. (2011) and Isidro et al. (2016). Hybrid metabolic models combining metabolic networks and partial least squares have been proposed by Ferreira et al. (2014) and Teixeira et al. (2011). The combination of systems of ODEs with neural networks (hybrid ODEs formalism) for the modeling of biochemical networks with intrinsic time delays has been studied by von Stosch et al. (2010). The integration of elementary flux modes (EMs) and PCA for hybrid metabolic pathway analysis has been researched by Folch-Fortuny et al. (2016) and von Stosch et al. (2016). Hybrid dynamic models that combine ODEs, PCA and EMs have been addressed by Folch-Fortuny et al. (2016). Lee et al. (2020) developed hybrid mechanistic/neural network models for partially known intracellular signaling pathways. Hybrid modeling approaches combining neural networks and ODEs have been applied to describe immunodeficiency virus (HIV) dynamics (2021) and coronavirus disease 2019 (COVID−19) dynamics (2020). Yang et al.

(2019) developed a white-box machine learning approach, leveraging carefully curated biological network models to mechanistically link input and output data, to reveal metabolic mechanisms of antibiotic lethality. Lewis and Kemp (2021) applied genome-scale flux balance analysis (FBA) to generate data to train ML classifiers to predict tumor radiosensitivity. Vijayakumar et al. (2020) developed a hybrid pipeline combining multi-omics ML with genome-scale FBA to analyze the phenotypic potential of *cyanobacterium*. Ramos et al. (2022) recently proposed a hybrid FBA technique that integrates GEMs and PCA constraints in a common linear program with mechanistic decision variables (fluxes) concomitantly with empirical decision variables (scores of principal components).

A large number of systems biology models, including GEMs, have been developed and stored in databases (e.g., BioModels (Le Noverre et al., 2006), JWS online (Olivier and Snoep, 2004), and KiMoSys (Mochao et al., 2020)) in the Systems Biology Markup Language (SBML) format (Hucka et al., 2003). SBML is a free and open standard based on XML to encode computational models of biological processes with widespread use in the systems biology scientific community. The SBML standard is, however, not commonly adopted in ML software tools. This significantly hinders the interlink between both modeling approaches in a hybrid workflow. Here, we propose a hybrid modeling framework that combines both modeling approaches and obeying the SBML standard. A previously published python package, SBML2HYB, is used to convert existing systems biology models into hybrid models and vice versa (Pinto et al., 2023). The so-formed hybrid models are trained with a deep learning algorithm based on ADAM, stochastic regularization and semidirect sensitivity equations (Pinto et al., 2022). The final (trained) hybrid models are uploaded in SBML databases, where they may be further analyzed as regular SBML models. This procedure was applied to three well-known models: the *E. coli* threonine pathway model (Chassagnole et al., 2001), the P58IPK signal transduction pathway model (Goodman et al., 2011) and the yeast glycolytic oscillations model (Dano et al., 2006).

6.2 Methods

6.2.1 General SBML Hybrid Model

SBML models are organized as j = 1, ..., n compartments with size V^j. Each compartment contains m^j species with a concentration vector c^j. The species are interlinked through q^j reactions with stoichiometry S^j and reaction kinetics r^j. SBML models also contain parame-

ters, θ, with given initial values (parameters may be local to reactions or global; for simplicity, we assume global). In SBML, the parameter values are not necessarily fixed as they may change over time according to predefined algebraic rules. The compartment size may also change over time according to predefined compartment rate rules (other rate rules were not considered here for simplicity). External time dependent stimuli may be defined through events, giving rise to a vector of exogenous input variables, u, that may change over time. With these elements, the dynamics of biochemical species in a generic compartment j may be described by the following ODEs model:

Equation 53. ODE model for an SBML model

$$\frac{d(c^j V^j)}{dt} = S^j \times r^j(c^j, \theta, u, \vartheta, t) \times V^j$$

$$\frac{dV^j}{dt} = z^j(V^j, c^j, \theta, u, \vartheta, t)$$

$$\theta = h(V^j, c^j, \theta, u, \vartheta, t)$$

Equation 53a is a conservation law of mass assuming a perfectly mixed compartment. Equation 53B represents a generic compartment rate rule in case the compartment size changes over time. Equation 53C represents generic algebraic rules to compute model parameters over time. Equation 53 is of a parametric nature with fixed structure stemming from prior knowledge (e.g., mass conservation laws, reaction stoichiometry or enzyme kinetics). Some variables may, however, lack a mechanistic basis (e.g., unknown reaction kinetics mechanisms or unknown physicochemical properties of molecular species such as charge or glycosylation pattern). In the general SBML hybrid model, variables lacking a mechanistic basis are defined as loose nonparametric functions, ϑ(·), without a fixed structure. They are computed by a deep feedforward neural network (FFNN) with nh hidden layers as a function of species concentrations, exogenous inputs, and other relevant variables (Equation 54):

Equation 54. General FFNN for an SBML model

$$H^0 = g(V^j, c^j, \theta, u, t)$$

$$H^i = \sigma\left(w^i \cdot H^{i-1} + b^i\right), \qquad i = 1, \ldots, nh$$

$$\vartheta(\cdot) = w^{nh+1} \cdot H^{nh} + b^{nh+1}$$

A non-linear pre-processing function, $g\,(V^j, cj, \theta, u, t)$, may be used to compute the FFNN input signals to improve the training. The input signals are forward propagated through the hidden layers. The σ(·) represents the nodes transfer function in the hidden layers (always the hyperbolic tangent function in this chapter). Finally, the FFNN outputs, ϑ, are computed by a linear output layer. The nodes connections weights, $w = \{w^1, w^2, \ldots, w^{nh+1}\}$ and $b =$

$\{b^1, b^2, \ldots, b^{nh+1}\}$, are calculated during the training of the model, for which an informative dataset is needed.

For a particular biological model, Equation 53 and Equation 54 describing n compartments with species and reactions are transformed via automatic symbolic manipulation into an equivalent set of ODEs and derived sensitivity equations using the Symbolic Math toolbox (MATLAB R2020a, MathWorks Inc.). The end result of this procedure is an automatically generated Matlab/Octave function that computes time derivatives of all state variables, $y = \{c^1, c^2, \ldots, c^n, V^1, V^2, \ldots, V^n\}$ (Equation 55):

Equation 55. General form of the automated derivatives function

$$\frac{dy}{dt} = f(y, \vartheta, u, t)$$

And also, the semidirect sensitivity parameters obtained by the symbolic differentiation of Equation 55 with respect to state variables, y, and FFNN outputs, ϑ (Equation 56):

Equation 56. General form of automated semidirect sensitivities

$$\frac{d\left(\frac{\partial y}{\partial \vartheta}\right)}{dt} = \left(\frac{\partial f}{\partial y}\right)\left(\frac{\partial y}{\partial \vartheta}\right) + \left(\frac{\partial f}{\partial \vartheta}\right)$$

$$\left(\frac{\partial y}{\partial \vartheta}\right)|_{t=0} = 0$$

Deep learning of hybrid models obeying to the system of Equation 55 and Equation 56 has been thoroughly investigated by Pinto et al. (2022). A Runge–Kutta 4th order ODE solver was implemented in MATLAB R2020a (MathWorks Inc.) to integrate the system of Equation 55 and Equation 56. The training was performed in a weighted least squares sense by minimizing the following loss function (Equation 57):

Equation 57. Loss function for an SBML hybrid model

$$WMSE = \frac{1}{T}\sum_{t=1}^{T}\frac{(y_t^* - y_t)^2}{\sigma_t^2}$$

with T the number of training examples, $y_t^*$the measured training example at time t, $y_t$the corresponding model prediction and σ_t the measurement standard deviation. The gradients of the loss function with respect to the neural network outputs were computed by the equation (Equation 58):

Equation 58. Loss function gradients

$$\frac{\partial WMSE}{\partial \vartheta} = -2 \sum_{t=1}^{T} \frac{y_t^* - y_t}{\sigma_i^2} \left(\frac{\partial y}{\partial \vartheta}\right)_t$$

The output layer gradients, $\frac{\partial WMSE}{\partial \vartheta}$, were backpropagated to the input layer via the well-known error backpropagation algorithm (Werbos, 1974), yielding the loss function gradients with respect to the neural network parameters (Equation 59):

Equation 59. Loss function gradients in respect to the FFNN parameters

$$g = \left[\frac{\partial WMSE}{\partial \omega}, \frac{\partial WMSE}{\partial b}\right]$$

Finally, the adaptive moment estimation algorithm (ADAM) (Kingma, 2014) with stochastic minibatch and weights dropout regularization was adopted to minimize the loss function given by Equation 57, using gradients, g (Equation 59). For further details, the reader is referred to Lee et al., 2020. The code was implemented in MATLAB R2020a (MathWorks Inc.) on a computer with Intel® Core™ i5–8265U CPU @ 1.60 GHz 1.80 GHz, and 24 GB of RAM.

6.2.2 Interfacing with SBML Databases and SBML Modeling Tools

The *SBML2HYB* python package (Pinto et al., 2023) was adopted to read SBML models, redesign them as hybrid models and to store them in model databases. This freely available python package converts existing systems biology models encoded in SBML into hybrid models that combine mechanistic equations and deep neural networks (currently limited to FFNNs). SBML is not a common format to encode ML models. An intermediate HMOD format supports the conversion process. The HMOD format is a text-based file (ASCII) with the list of properties defining the model (species, reactions, parameters, rates, and rules) in a similar manner to SBML, by considering any number of species with a certain initial concentration distributed among any number of compartments. These species are then interlinked through a list of reactions and rate rules. The user inputs the information of the deep neural network into the HMOD file either manually or through a pre-configured neural network in Python *keras,* using the SBML2HYB tool. The resulting hybrid model in HMOD format is reconverted to SBML and uploaded in model databases. In this step, the FFNN Equation 54 is mapped to assignment rules in SBML format, whereas the network weights are mapped to global parameters in the SBML format. The resulting SBML hybrid models may be simulated, analyzed and/or trained with existing tools such as MATLAB (MathWorks Inc.), COPASI (Hoops et al.,

2006) or special purpose tools with training algorithms for hybrid models that are able to read SBML files. For further details, the reader is referred to (Pinto et al., 2023).

6.2.3 Case Studies

The SBML hybrid modeling framework was applied to three systems biology case studies freely available in the JWS Online database (https://jjj.bio.vu.nl/models/, accessed on January 2024) (Olivier and Snoep, 2004) with the access IDs given in Table 12. The first case study is a metabolic network describing the synthesis of threonine in *E. coli* proposed by Chassagnole et al. (2001). The second case study is the P58IPK signal transduction network to study *Influenza* infection dynamics proposed by Goodman et al. (2011). The third case study is a reduced yeast glycolytic model with preserved limit cycle stability proposed by Dano et al. (2006). In order to upgrade the original mechanistic models in hybrid mechanistic/neural network versions, the following pipeline of activities (Figure 23) was applied to each of the case studies:

Step 1: The original systems biology models were retrieved from the JWS database in SBML format. The respective files are provided as supplementary material.

Step 2: Synthetic time series datasets were generated by simulating the original models in the JWS platform. The resulting data sets are provided as supplementary material. These data are needed to train the hybrid models as a proof-of-concept. No experimental data were used in this study. More details are provided in section 6.3.

Step 3: For each case study, a feedforward neural network (FFNN) was inserted into the mechanistic model and converted to the HMOD format using the *SBML2HYB* python tool, freely available in Pinto et al., 2023. The size of the FFNN and interface with the mechanistic model depended on the case study. More details are given in section 6.3.

Step 4: The hybrid mechanistic/FFNN models encoded in the HMOD format were trained using the deep learning approach described in Section 2.1 and the datasets generated in step 2. Implementation details varied in the case studies (more on this in section 6.3). The main concern was the proof-of-concept that SBML hybrid models may be efficiently trained to a comparable performance to the original mechanistic models. The effect of the size of the FFNN was investigated. The final trained hybrid models, with the updated FFNN weights, were saved in the HMOD format.

Step 5: The trained hybrid models in the HMOD format were reconverted to SBML using the *SBML2HYB* tool. In this step, the FFNN information is mapped to assignment rules in the SBML format. The obtained SBML files were uploaded to the JWS online platform and are

now freely available for the community to analyze. The hybrid model structures encoded in SBML were visualized using the freely available Cytoscape cy3sbml tool (Konig et al., 2012). The hybrid models SBML files are provided as supplementary material.

Step 6: For proof-of-concept, the original mechanistic SBML models (step 1) and the final hybrid SBML models (step 5) were simulated and compared using the JWS online simulator (https://jjj.bio.vu.nl/models/experiments/, accessed on January 2024) showing that their outputs are practically coincident.

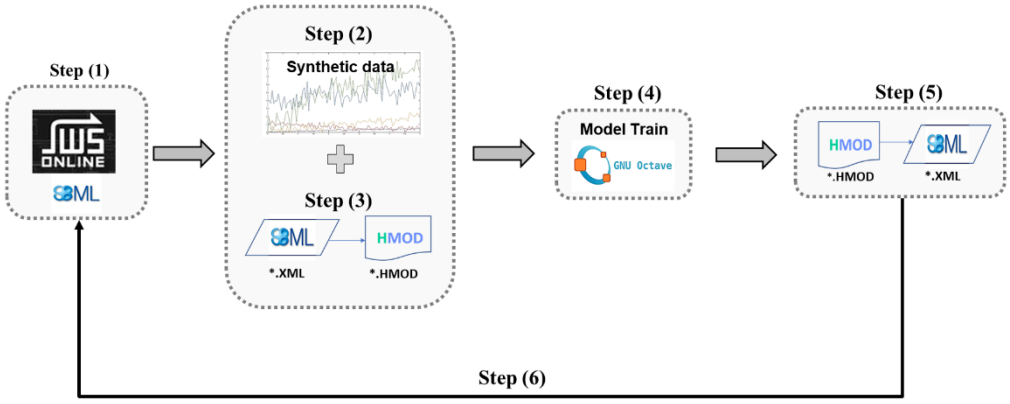

Figure 23. Schematic workflow for redesigning existing SBML models stored in databases into hybrid mechanistic/neural network models. Step 1: An SBML biologic model is extracted from a model database. Step 2: A synthetic time series dataset is generated to train the hybrid model. Step 3: A feedforward neural network (FFNN) is inserted in the mechanistic kinetic model and converted to the HMOD format using the SBML2HYB tool. Step 4: The hybrid mechanistic/FFNN model encoded in the HMOD format is trained by applying the deep learning approach (Section 6.2.1) and the synthetic dataset. Step 5: The trained hybrid model in the HMOD format is reconverted to SBML using the SBML2HYB tool. Step 6: The final trained hybrid model in the SBML format is uploaded in the model database and simulated comparatively to the original nonhybrid model.

Table 12. Summary of the three SBML models that were redesigned to hybrid mechanistic/neural network models in the present study.

Case Study	Number of Species	Number of Reactions	Number of Parameters	JWS Online ID	Reference
E. coli threonine synthesis pathway	11	7	47	chassagnole1	Chassagnole et al., 2001
P58IPK signal transduction pathway	9 (4 fixed)	9	10	goodman	Goodman et al., 2011
Yeast glycolytic oscillations	7 (1 fixed)	11	31	dano1	Dano et al., 2006

As mentioned in step 5, hybrid models with different network depths and sizes were evaluated for each case study. The "best" hybrid model was discriminated on the basis of the Akaike Information Criterion with a second order bias correction (*AICc*), computed for the training data partition as follows (Equation 60):

Equation 60. Akaike Information Criterion with second order bias correction

$$AICc = T\ ln\ +\ 2\ nw + \frac{2\ nw\ (nw\ +\ 1)}{T\ -\ nw\ -\ 1}$$

With nw the total number of FFNN weights that are calculated during the training process. *AICc* includes an overparameterization penalty and is commonly used to discriminate between empirical model candidates and to select a parsimonious model for small sample sizes (Li et al., 2002).

6.3 Results and Discussion

6.3.1 Case Study 1: Threonine Synthesis Pathway in E. coli

The first case study is the metabolic model proposed by Chassagnole et al. (2001), describing the threonine synthesis pathway in *E. coli* (Table 12). This model dynamically simulates the time course of 11 species (adp, asa, asp, aspp, atp, hs, hsp, nadp, naph, phos and thr) in a single compartment, corresponding to 11 ODEs. It has seven reactions (with rates vak, vasd, vatpase, vhdh, vhk, vnadph_endo and vtsy) and 47 kinetic parameters (the names of variables were kept the same as in the original SBML model to facilitate cross-reference; for details, the reader is referred to the JWS Online model with access ID 'chassagnole').

Hybrid models were created by combining deep FFNNs of different sizes with the original mechanistic model, following the previously described procedure (Figure 23). The FFNNs had 11 inputs corresponding to the concentrations of the 11 species (adp, asa, asp, aspp, atp, hs, hsp, nadp, naph, phos, thr). The number of hidden layers and nodes in the hidden layers varied (Table 2). The activation function in the hidden layers was always the hyperbolic tangent function. The FFNNs had seven outputs corresponding to the maximum reaction rate values of the seven metabolic reactions. The kinetic equations of the original SBML model were fully kept in the hybrid models. The job of the FFNNs was thus to describe the maximum reaction rate parameters as a function of species concentrations. Figure 24 graphically represents the hybrid model structure [11 × 5 × 5 × 7] (Table 13) using the Cytoscape cy3sbml tool. This figure shows an heterogenous (hybrid) network composed of nodes and edges of different

nature. On the biochemical network side (left), the large circles represent the molecular species, which have a physical concentration associated. The small black squares and respective edges represent biochemical reactions with a well-defined stoichiometry. The black triangles are the reaction kinetic rates. On the feedforward neural network side (right), the blue circles represent the neural network nodes, which have an abstract numerical value associated defining the node strength. The green squares and respective edges represent signal propagation between nodes. The interlink between the two sides of the network is mediated by the black triangles, which in this case correspond to the maximum reaction rate parameters to be applied in the kinetic law equations. An interesting analogy may be established between the neural network part and an artificial nucleus of a cell with associated signal transduction networks and gene regulatory networks, with the job of controlling the underlying metabolic processes.

Table 13. Training metrics of different hybrid models for the *E. coli* threonine synthesis pathway case study (chassagnole1). The dataset was divided in four experiments for training (400 training examples for each state variable) and five for testing (500 testing examples for each state variable). The training was performed with ADAM with default hyperparameters as suggested by Kingma (2014) (α=0.001, =0.9, =0.999 and $\zeta = 1 \times 10^{-8}$). The number of iterations was 5000. The minibatch size was 78% and weight dropout probability was 0.22 as suggested by Pinto et al. (2022). The AICc was computed on the training set only. The noise-free WSSE measures the error between noise-free data (e.g., true process behavior) and model predictions.

Hybrid model	WMSE Train	WMSE Test	WMSE Test (Noise Free)	AICc	CPU Time (hh:mm:ss)	Number of Weights
11 × 5 × 5 × 7	1.03	0.99	0.07	838	00:31:00	132
11 × 10 × 10 × 7	1.07	1.00	0.08	2510	00:29:00	307
11 × 15 × 15 × 7	1.04	0.99	0.08	2102	00:35:00	532
11 × 20 × 20 × 7	1.03	0.98	0.07	2400	00:33:00	807
11 × 5 × 5 × 5 × 7	1.03	0.99	0.07	918	00:32:00	162
11 × 10 × 10 × 10 × 7	1.05	0.98	0.07	1890	00:40:00	417
11 × 15 × 15 × 15 × 7	1.04	1.01	0.08	2659	00:36:00	772
11 × 20 × 20 × 20 × 7	1.04	1.00	0.07	3684	00:35:00	1227

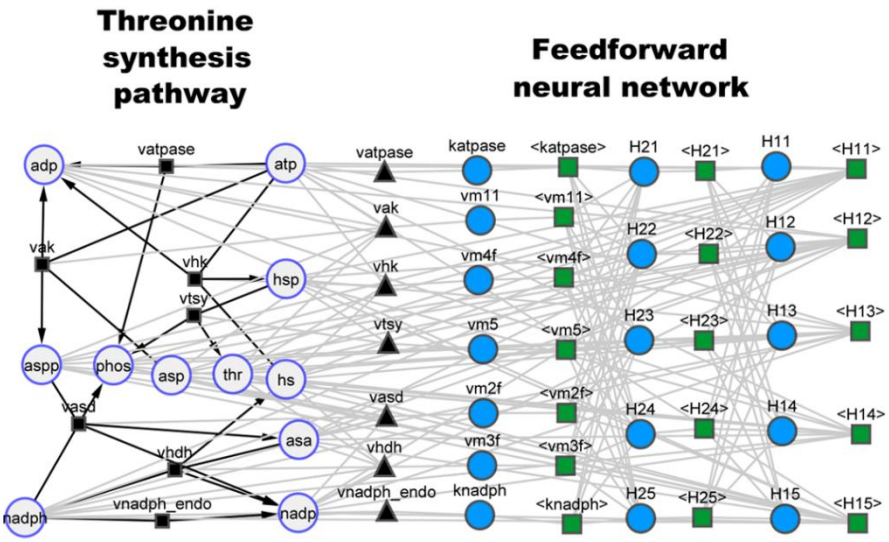

Figure 24. Hybrid model structure [11 × 5 × 5 × 7] for the threonine synthesis pathway (1st row of Table 2) visualized in the cy3sbml tool (Konig et al., 2012). **Left side**: Metabolic network with physical meaning. Large circles represent biochemical species (metabolites). Black squares and black edges represent biochemical reactions. Black triangles represent kinetic laws. **Right side:** Artificial feedforward neural network with size [11 × 5 × 5 × 7]. Small blue circles represent neural network nodes. Green squares and gray edges represent signal propagation between neural network nodes. The first layer receives input signals of biochemical species concentrations (Large circles). The last layer delivers kinetic parameter values to the black triangles, which mediate the communication between both sides of the network.

The hybrid models were trained with a synthetic data set following the procedure of Figure 23. A time series dataset was created by simulating the original SBML model directly in the JWS platform. A two-factor central composite design of experiments (CC-DOE) was carried out to the initial concentrations of atp between 5 and 15 (arbitrary units) and of asp between 1 and 3 (arbitrary units) resulting in nine experiments. The data for each experiment was recorded as a time series with 100 data points and a sampling time of 1 (arbitrary units). Gaussian noise (10%) was added to concentrations of species, thereby simulating experimental error. This synthetic dataset is available in the supplementary material (Simulation_data.xlsx; chassagnole_data sheet). From the nine experiments, four were used for training (the star experiments of the CC-DOE corresponding to 400 training examples for each state variable) and five were used for testing (the square plus the center experiments of the CC-DOE corresponding to 500 training examples for each state variable). The training was performed with ADAM with default hyperparameters (Table 13), 5000 iterations, semidirect sensitivity equations and stochastic regularization with a minibatch size of 0.78 and weights dropout of 0.22. The choice of the minibatch size and weights dropout was based on the results by Pinto et al.

(2022). Table 13 shows the overall training metrics for different FFNNs sizes. The performances of the hybrid models in terms of training error (WMSE train) and testing error (WMSE test) are comparable. The magnitude of the train and test errors are also comparable, denoting an effective training without overfitting in all cases. This is further strengthened by the very low noise-free test error showing that model predictions are very close to the true process behavior in all cases. The total number of network weights varied almost 10-fold, but this was not reflected in the training performance. The best hybrid structure was chosen to be the smallest one [11 × 5 × 5 × 7] based on the lowest AICc value (1st row in Table 13).

The trained hybrid models may be simulated and analyzed in any systems biology platform complying with the SBML standard. As proof-of-concept, the best hybrid model [11 × 5 × 5 × 7] in the SBML format was uploaded to the JWS online platform and simulated. Figure 25 shows the JWS online simulation of the original model and of the best hybrid model [11 × 5 × 5 × 7] for a test experiment not used for training (the center point experiment of the CC-DOE). The results show that the hybrid model perfectly mimicked the dynamics of the original mechanistic model.

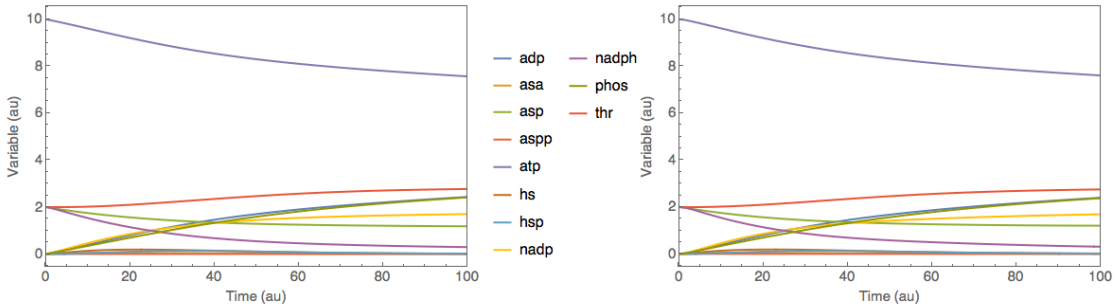

Figure 25. Comparison between original model and best hybrid model for case study 1 (threonine synthesis pathway in *E. coli*) Dynamic profiles were simulated based on the respective SBML files in the JWS Online platform. The test experiment was the center point experiment of the CC-DOE (not used for training). Full lines represent species concentrations over time. Left panel: Original SBML model simulation. Right panel: Best hybrid model simulation with structure [11 × 5 × 5 × 7] (First row of Table 13).

The procedure presented in Figure 23 may result in mathematical structures that are more detailed mechanistically and much more complex to train than previously published hybrid models. This may raise concerns about the training feasibility of FFNNs interlinked with complex mathematical structures. Pinto et al. (2022) compared traditional shallow hybrid modeling (using the Levenberg–Marquardt algorithm coupled with the indirect sensitivity equations, cross-validation, and a hyperbolic tangent activation function) with deep hybrid modeling (using ADAM, semidirect sensitivity equations, stochastic regularization and multiple hidden layers). A clear advantage of hybrid deep learning both in terms of predictive power and

computational cost was demonstrated. However, all experiments had a simplistic mechanistic part. Here, case study 1 model kept the original kinetic law equations. Seven highly complex kinetic equations with 47 parameters were "merged" with the FFNN. Table 13 results suggest nonetheless that the previously published deep learning approach for hybrid models (ADAM + semidirect sensitivity equations + stochastic regularization) is equally effective at training hybrid models with complex parametric functions.

6.3.2 Case Study 2: P58IPK Signal Transduction Pathway

The second case study was based on the viral infection model proposed by Goodman et al. (2011), freely available in SBML in the JWS Online database (http://www.jjj.bio.vu.nl, accessed on January 2024) under access ID 'goodman' (Table 12). The authors studied the dynamics of the P58IPK signal transduction pathway during *Influenza* virus infection. A mathematical model was developed to evaluate the effect of protein P58a activation on the P58IPK pathway dynamics, particularly on the activation of the PKR kinase and on the phosphorylation of eIF2, both controlling viral protein expression. The model comprehends nine species (Flu, NS1, P58a, P58total, PKRp, PKRtotal, eIF2ap, eIF2atotal and ext) in a single compartment, of which four are fixed (P48total, PKRtotal, eIF2atotal and ext), corresponding to five ODEs. The model further has nine reactions and 10 parameters. The names of variables were kept the same as in the original model and are explained in the database.

As in the previous case study, SBML hybrid models were created by combining FFNNs of different sizes (Table 14) with the original mechanistic model following the procedure of Figure 23. Figure 26 shows the hybrid model structure [5 × 10 × 10 × 10 × 9] (Table 14) using the SBML-visualizing cy3sbml tool (Konig et al., 2012. The left side of Figure 26 represents the original mechanistic signal transduction network, whereas the right side represents the FFNN added to the mechanistic core. The FFNN has five inputs corresponding to the concentrations of the five dynamical species (Flu, NS1, P58a, PKRp and EIF2ap), three hidden layers (10 × 10 × 10) with hyperbolic tangent activation functions, and nine outputs corresponding to the kinetic rates (v_1r, v_2r, v_3r, v_4r, v_5r, v_6r, v_7r, v_8r, v_9r as they are named in the original SBML implementation). In this case study, the FFNNs in Table 14 completely replaced the kinetic laws of the original model, which were therefore deleted in the hybrid model structures. This network may be interpreted as a hybrid signal transduction pathway with a physical part composed of proteins and an artificial part composed of abstract neural network nodes.

Table 14. Training metrics of different hybrid models for the P58IPK signal transduction pathway case study (goodman). The dataset was divided into four experiments for training (400 training examples for each state variable) and five for testing (500 testing examples for each state variable). The training was performed with ADAM with default hyperparameters as suggested by Kingma (2014) (α = 0.001, = 0.9, = 0.999 and $\zeta = 1 \times 10^{-8}$). The number of iterations was 5000. The minibatch size was 78% and weight dropout probability was 0.22 as suggested by Pinto et al. (2022). The AICc was computed on the training set only. The noise-free WSSE measures the error between noise-free data (e.g., true process behavior) and model predictions.

Hybrid model	WMSE Train	WMSE Test	WMSE Test (Noise Free)	AICc	CPU Time (h:m:s)	Number of Weights
5 × 5 × 5 × 9	1.60	1.51	0.54	1916	00:12:10	114
5 × 10 × 10 × 9	1.59	1.48	0.53	2181	00:11:54	269
5 × 15 × 15 × 9	1.61	1.50	0.56	2810	00:15:15	474
5 × 20 × 20 × 9	1.58	1.49	0.51	3480	00:20:48	729
5 × 5 × 5 × 5 × 9	1.45	1.50	0.48	1890	00:13:15	144
5 × 10 × 10 × 10 × 9	1.23	1.28	0.12	1430	00:16:10	379
5 × 15 × 15 × 15 × 9	1.35	1.36	0.31	2140	00:19:30	714
5 × 20 × 20 × 20 × 9	1.34	1.40	0.36	4150	00:27:12	1149

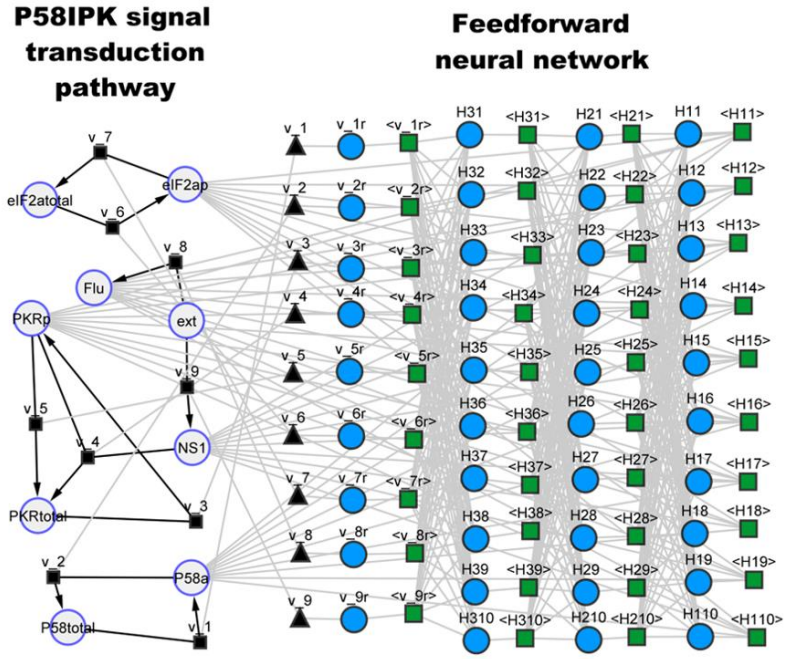

Figure 26. Hybrid model structure [5 × 10 × 10 × 10 × 9] for the P58IPK signal transduction pathway (6th row of Table 3) visualized using the cy3sbml tool (Konig et al., 2012). **Left side**: Signal transduction network with physical meaning. Large circles represent biochemical species (proteins). Black squares and black edges represent biochemical reactions. Black triangles represent kinetic laws. **Right side**: Artificial feedforward neural network with size [5 × 10 × 10 × 10 × 9]. Small blue circles represent neural network nodes. Green squares and gray edges repre-

sent signal propagation between neural network nodes. The first layer receives input signals of biochemical species concentrations (Large circles). The last layer delivers kinetic parameter values to the black triangles, which mediate the communication between both sides of the network.

Hybrid SBML models with varying number of hidden layers and nodes in the hidden layers were trained using a synthetic data set. A time-series dataset was created by simulating the original SBML model in the JWS platform following a similar procedure to case study 1 (available in the supplementary material as Simulation_data.xlsx; goodman_data sheet). A two-factor CC-DOE was carried out to the initial amount of Flu (overall level of infection within the host cell) between 2 and 6 (arbitrary units) and the initial amount of PKRp (phosphorylated PKR protein) between 0 and 2 (arbitrary units). The data for each experiment were recorded as a time series with 100 points and sampling time of 0.05 (arbitrary units). This resulted in nine experiments with 100 time points each. Additionally, 10% Gaussian noise was added to concentrations of species to simulate experimental error. As in the previous case study, four experiments were selected for training (the star experiments of the CC-DOE corresponding to 400 training examples for each state variable), and five experiments were used for testing (the square plus the center experiments of the CC-DOE corresponding to 500 training examples for each state variable). The training was performed using ADAM with default hyperparameters (Table 14), 5000 iterations, semidirect sensitivity equations and stochastic regularization (minibatch size of 0.78 and weight dropout of 0.22, as before). The overall training results for different FFNN sizes are shown in Table 14. As opposed to the previous case study, the size of the FFNN has an effect on the training performance. This may be explained by the smaller amount of mechanistic knowledge embodied in the hybrid models. Since the original kinetic laws were completely deleted in the hybrid models, the training results are more heavily dependent on the FFNN structure. Interestingly, the larger networks with a higher depth (three hidden layers) outperformed the smaller networks, particularly in the extrapolation experiments (test WMSE). Overall, the structure [5 × 10 × 10 × 10 × 9] stands out as the best-performing model with the lowest training error (WMSE train) and lowest testing error (WMSE test). This is further reinforced by the lowest noise-free test error and the lowest AICc. This structure was uploaded to the JWS online platform and simulated comparatively to the original mechanistic model (Figure 27). As in the previous case study, the best-performing hybrid SBML model [5 × 10 × 10 × 10 × 9] was able to perfectly mimic the dynamics of the original mechanistic model.

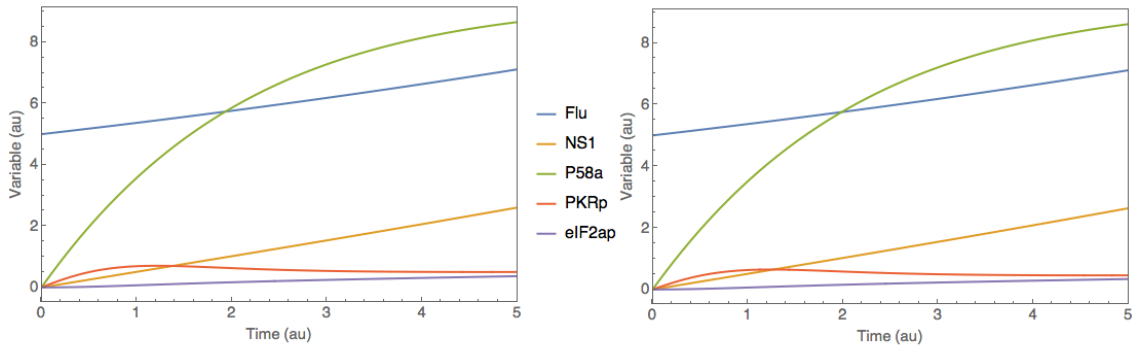

Figure 27. Comparison between original model and best hybrid model for case study 2 (P58IPK signal transduction pathway). Dynamic profiles were simulated based on the respective SBML files in the JWS Online platform. The test experiment was the center point experiment of the CC-DOE (not used for training). Full lines represent species concentrations over time. Left panel: Original SBML model simulation. Right panel: Best hybrid model simulation with structure [5 × 10 × 10 × 10 × 9] (Sixth row of Table 14).

6.3.3 Case Study 3: Yeast Glycolytic Oscillations

The third case study consisted of the reduced dynamical model of yeast glycolysis proposed by Dano et al. (2006). This model is a reduced version of a more detailed yeast glycolysis model. Both the original and reduced models exhibit limit cycle stability, with a certain number of species showing stable oscillations over time. It comprehends eight species (ADP, AMP, ATP, BPG, DHAP, FBP, GAP and sink) in a single compartment. The dynamic variable 'sink' was the only one that was fixed, thus translating to a system of seven ODEs. The model further comprehends 11 metabolic reactions and 31 parameters. This model is freely available in SBML format on the JWS Online database (http://www.jjj.bio.vu.nl, accessed on January 2024) with access ID 'dano1'.

SBML hybrid models were created by combining FFNNs of different sizes with the original mechanistic model. Figure 6 illustrates this process for the structure [7 × 10 × 10 × 10 × 11] with 421 weights (6th row of Table 15). The right side of Figure 28 represents the original metabolic network, whereas the left side represents the incorporated FFNN. In this example, the FFNN has seven inputs corresponding to the concentrations of the seven species (ADP, AMP, ATP, BPG, DHAP, FBP, GAP), three hidden layers (10 × 10 × 10) with hyperbolic tangent activation functions, and 11 outputs corresponding to the kinetic rates (v_1r, v_2r, v_3r, v_4r, v_5r, v_6r, v_7r, v_8r, v_9r, v_10r, v_11r as they are named in the original SBML model). As in case study 2, the original kinetic laws were completely deleted in the hybrid models.

Table 15. Training metrics of different hybrid models for the yeast glycolytic oscillations case study (Dano1). The dataset was divided into four experiments for training (400 training examples for each state variable) and five for testing (500 testing examples for each state variable). The training was performed with ADAM with default hy-

perparameters as suggested by Kingma (2014) (α = 0.001, = 0.9, = 0.999 and $\zeta = 1 \times 10^{-8}$). The number of iterations was 10000. The minibatch size was 78% and weight dropout probability was 0.22 as suggested by Pinto et al. (2022). The AICc was computed on the training set only. The noise-free WSSE measures the error between noise-free data (*e.g.,* true process behavior) and model predictions.

Hybrid model	WMSE Train	WMSE Test	WMSE Test (Noise Free)	AICc	CPU time (h:m:s)	Number of Weights
7 × 5 × 5 × 11	20.12	21.05	20.14	5730	01:05:00	136
7 × 10 × 10 × 11	1.87	1.99	1.67	3818	01:20:00	311
7 × 15 × 15 × 11	1.74	1.78	1.56	4120	01:15:00	536
7 × 20 × 20 × 11	1.16	1.43	0.98	2740	01:24:00	811
7 × 5 × 5 × 5 × 11	5.33	5.84	5.14	3930	01:33:00	166
7 × 10 × 10 × 10 × 11	0.93	0.94	0.11	−41	01:31:00	421
7 × 15 × 15 × 15 × 11	0.98	0.97	0.21	784	01:20:00	776
7 × 20 × 20 × 20 × 11	0.97	0.97	0.17	2213	01:40:00	1231

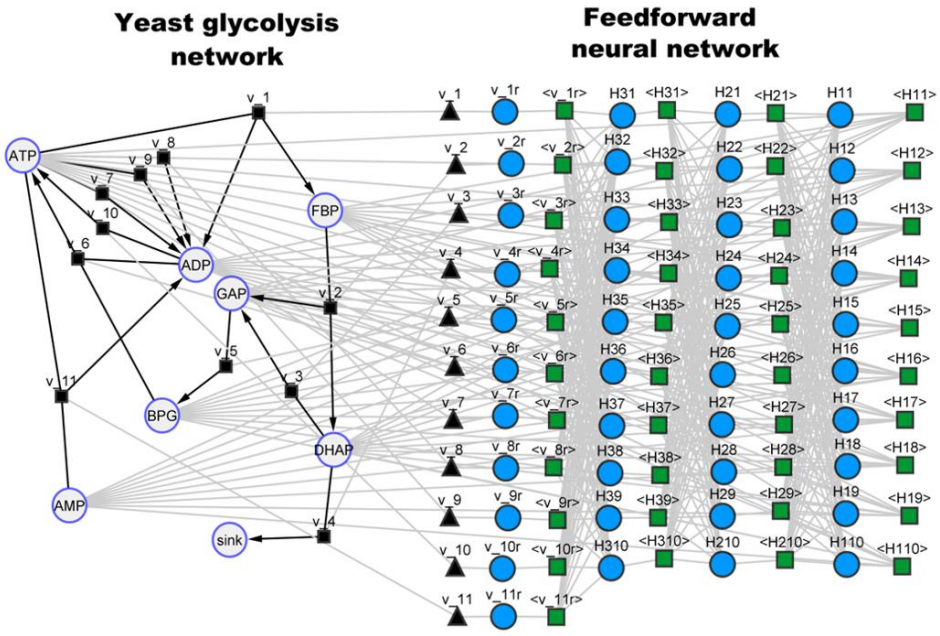

Figure 28. Hybrid model structure [7 × 10 × 10 × 10 × 11] for the yeast glycolysis pathway (6th row of Table 15) visualized in the cy3sbml tool (Konig et al., 2012). **Left side**: Reduced glycolysis network with physical meaning. Large circles represent biochemical species (metabolites). Black squares and black edges represent biochemical reactions. Black triangles represent kinetic laws. **Right side**: Artificial feedforward neural network with size [7 × 10 × 10 × 10 × 11]. Small blue circles represent neural network nodes. Green squares and gray edges represent signal propagation between neural network nodes. The first layer receives input signals of biochemical species concentrations (Large circles). The last layer delivers kinetic parameter values to the black triangles, which mediate the communication between both sides of the network.

Hybrid SBML models of different sizes were trained with a synthetic dataset following a similar process to the previous case studies. A two-factor CC-DOE was carried out by varying the amount of initial ADP concentration between 1 and 2 (arbitrary units) and the initial ATP concentration between 1 and 2 (arbitrary units), resulting in nine experiments. Each experiment was simulated on the JWS Online platform with the resulting time-series data (100 time points) recorded with a sampling time of 0.05 (arbitrary units). Gaussian noise (10%) was added to the concentrations of species. This synthetic dataset is available as supplementary material (Simulation_data.xlsx; dano1_data sheet). Four experiments were selected for training (the star experiments of the CC-DOE corresponding to 400 training examples for each state variable) and five were used for testing (the square plus the center experiments of the CC-DOE corresponding to 500 training examples for each state variable). The hybrid models were trained with this data using ADAM with default hyperparameters (10000 iterations, semidirect sensitivity equations, stochastic regularization with minibatch size of 0.78 and weight

dropout of 0.22). The overall training results for different FFNNs sizes are shown in Table 15. Unsurprisingly, limit cycle stability is a more challenging problem for hybrid model development. The effect of the FFNN depth and size was much more pronounced than in the previous example. The smaller networks were not able to exhibit stable oscillations even for the training examples. Only models with three hidden layers were able to accurately capture the oscillatory dynamics. The three largest structures show a comparable training and testing error. However, the structure [7 × 10 × 10 × 10 × 11] clearly stands out as the best-performing model with the lowest training error (WMSE train) and the lowest testing error (WMSE test). This is further accentuated by the significantly lower noise-free test error and lower AICc. This hybrid SBML model was uploaded to the JWS online platform and simulated comparatively to the original metabolic model for the center point test experiment (not used for training) of the CC-DOE (Figure 29). Remarkably, the best hybrid model structure [7 × 10 × 10 × 10 × 11] was able to reproduce very faithfully the oscillatory behavior of the original metabolic model when exposed to different initial conditions than those applied in the training experiments.

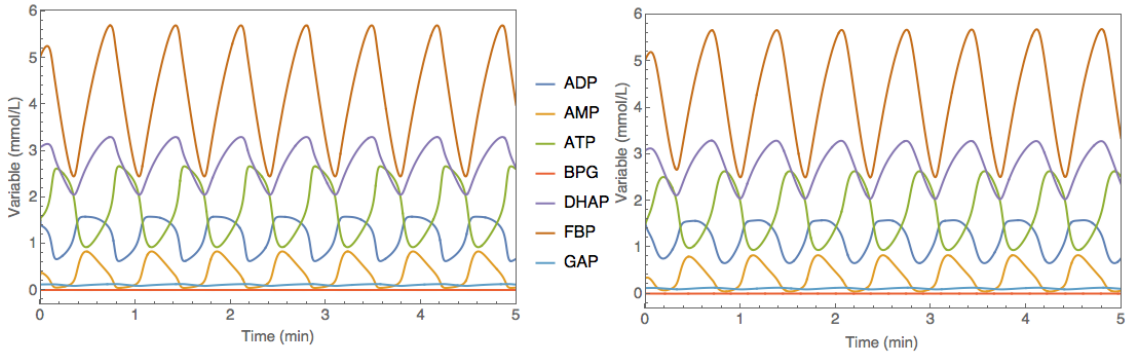

Figure 29. Comparison between original model and best hybrid model for case study 3 (yeast glycolysis model). Dynamic profiles were simulated based on the respective SBML files on the JWS Online platform. Simulations were performed for the center point experiment of the CC-DOE (not used for training). Full lines represent species concentrations over time. Left panel: Original SBML model simulation. Right panel: Best hybrid model simulation with structure [7 × 10 × 10 × 10 × 11] (Sixth row of Table 15).

6.4 Conclusions

SBML is an open standard based on XML currently adopted by the systems biology community to encode computational models of biological processes. An extensive body of research has produced a large number of such SBML models that are currently stored in public databases. The SBML standard is, however, not commonly adopted to encode ML models. The main novelty of the present study is the combination of both modeling formalisms in a

common hybrid workflow obeying the SBML standard. With few exceptions, previously published hybrid models embodied relatively simple mechanistic models (mechanistic scale-gap) and relatively simple ML models (ML scale-gap). With the proposed SBML hybrid modeling framework, the mechanistic scale-gap may be significantly narrowed. It is shown with three simple examples how publicly available SBML models may be easily upgraded to hybrid mechanistic/neural network models obeying the SBML standard. Such hybrid models may be trained with state-of-the-art deep learning algorithms to either mimic, improve or extend existing SBML models. They may be further uploaded, trained, and analyzed in SBML compatible software tools. Even if the presented examples are relatively simple, the proposed framework is, in principle, directly scalable to larger whole organism models, eventually at the genome-scale. All in all, it is expected this framework to greatly facilitate the adoption of hybrid mechanistic/ML techniques to develop computational models of biological systems.

7

CONCLUSIONS AND FUTURE WORK

7.1 Conclusions

The main objectives of this PhD dissertation was to develop a deep hybrid modelling methodology that combines mechanistic models with emergent deep neural networks and the implementation of these developed hybrid modelling methods in a way that is scalable to large Systems Biology models and Systems Biology Markup Language (SBML) compatible.

To fulfill the general objective, in Chapter 3 the general bioreactor hybrid model was revisited and the use of deep learning techniques in the context of hybrid modeling. Two different approaches were applied, and their results compared: First, the traditional approach using Levenberg-Marquardt optimization coupled with the indirect sensitivities, cross-validation, and *tanh* activation function. Second, the novel hybrid deep approach that uses the adaptive moment estimation method (ADAM), semidirect sensitivities, stochastic regularization and *ReLU* activation functions in the hidden layers. Overall, the results showed that the deep learning method has better predictive capabilities along with some other advantages: First, it is practically insensitive to weight initialization thereby eliminating the need for training repetitions. Second, the stochastic nature of the method is less sensitive to experimental noise, eliminating the need for cross-validation. Lastly, the introduction of semidirect sensitives further decreases the CPU time particularly for large deep structures as the number of sensitivity equations (that need to be integrated over time) becomes independent of the number of hidden layers.

In Chapter 4, a hybrid modeling framework that makes use of deep learning with state-space reduction was applied for data analysis and design space exploration to a case of *P. pastoris* GS115 (Mut+) cultures expressing a scFv. The state-space reduction consisted in using a PCA

in order to reduce the number of species requiring time series integration from 9 to a lower amount (optimal values were achieved with 5 principal components). In this scenario, the inorganic elements in the medium can have a large impact on the kinetics. The state-space reduction framework succeeded in decreasing the model complexity by 60% and improving the predictive power by 18.5% in relation to a standard nonreduced hybrid model. The reduced hybrid model was able to correctly simulate the experiments performed including the test experiments. It should be noted that, despite the success of this approach, more data is required to strengthen model validation before it can be considered for a process digital twin.

In Chapter 5, the first comparison between the deep and shallow hybrid modeling approaches on a CHO-K1 fed-batch process in a process development campaign was carried out. In this chapter 2 case studies were used. The first was a synthetic dataset that considered the existence of 25 extracellular species (considered to be "measured" for model training purposes) and 21 intracellular species (considered to be "unmeasured" and, as such, hidden from the model during training). The second was an experimental dataset with 30 measured species. Of note, the main challenges of each were, for the first case, the existence of hidden states (the 21 intracellular species) and, for the second case, the switch between lactate production and lactate consumption as well as from ammonium production and ammonium consumption. The obtained results pointed to a systematic improvement in the generalization capabilities of the model when using a deep hybrid model in comparison to the shallow hybrid model, including successfully solving the main challenges identified. These results are in line with the generally accepted view that deep neural networks have a better generalization power than shallow networks. This chapter points to similar conclusions when dealing in a hybrid modeling context.

Lastly, Chapter 6 introduced SBML compatibility to the hybrid modeling paradigm. A methodology was proposed that allows an SBML model to be hybridized or for a hybrid SBML compatible model to be created from the start. The proposed framework allows, as such, for a significant reduction in the mechanistic scale-gap. This framework was tested with three case studies, starting at a mechanistic model from a database, those models were turned into hybrid models, trained, and then reuploaded to the database. All the case studies were successful in creating a hybrid model with results comparable to the original mechanistic model (which was the training objective). All in all, it is hoped that this framework will greatly facilitate the adoption of hybrid mechanistic/ML techniques in the development of computational models of biological systems.

Overall, this thesis proposes a hybrid modelling framework that makes use of state-of-the-art deep learning techniques to improve upon the classic shallow hybrid models. Furthermore, the proposed framework is shown to be SBML compatible. The results show an all-around improvement in the predictive capabilities of the models generated with this framework when compared to the classic approach. The SBML compatibility can also facilitate the dissemination of hybrid models in the Systems Biology community.

7.2 Future Work

Only recently has the hybrid modelling community started the shift from shallow non-parametric parts (mostly non-deep FFNNs) to deep ones. This shift comes at a time when, with the recent explosion in the field of Artificial Intelligence (AI) methods, it can be expected that new approaches will keep appearing and have a further impact in how hybrid modelling is conducted. As deep hybrid modelling techniques have shown to be an overall improvement over their shallow counterparts, future work should be directed towards applying deep learning centric approaches from the AI fields to the hybrid paradigm.

Among these, some novelties that can be of high interest for hybrid modelling are the use of Physics Informed Neural Networks (PINNs) and Long Short-Term Memory (LSTM) networks. Further work should also be done in scaling up the sizes of the hybrid models (i.e. Genome Scale models) to allow for increasingly complex scenarios to be described with relative ease.

Lastly, the application of hybrid models to real time process control and optimization is also an area with large interest as it can reduce the amounts of waste/increase productivity in bioprocess development, such as the highly competitive and highly complex biopharma industry.

REFERENCES

Abonyi, J., Chovan, T., Nagy, L., & Szeifert, F. (1999). Hybrid convolution model and its application in predictive pH control. Computers and Chemical Engineering, 23, S227–S230.

Agharafeie, R., Ramos, J. R. C., Mendes, J. M., & Oliveira, R. (2023). From Shallow to Deep Bioprocess Hybrid Modeling: Advances and Future Perspectives. *Fermentation*, *9*(10), 922.

Aguiar, H. C., & Maciel Filho, R. (2001). Neural network and hybrid model: a discussion about different modeling techniques to predict pulping degree with industrial data. *Chemical engineering science*, *56*(2), 565-570.

Alpaydin, E. (2020). *Introduction to machine learning*. MIT press.

Al-Yemni, M., & Yang, R. Y. (2005). Hybrid neural-networks modeling of an enzymatic membrane reactor. *Journal of the Chinese Institute of Engineers*, *28*(7), 1061-1067.

Anderson, E. C., Bell, G. I., Petersen, D. F., & Tobey, R. A. (1969). Cell growth and division: IV. Determination of volume growth rate and division probability. *Biophysical journal*, *9*(2), 246-263.

Anderson, J. S., McAvoy, T. J., & Hao, O. J. (2000). Use of hybrid models in wastewater systems. *Industrial & Engineering Chemistry Research*, *39*(6), 1694-1704.

Andrews, J. F. (1968). A mathematical model for the continuous culture of microorganisms utilizing inhibitory substrates. *Biotechnology and bioengineering*, *10*(6), 707-723.

Antonakoudis, A., Barbosa, R., Kotidis, P., & Kontoravdi, C. (2020). The era of big data: Genome-scale modelling meets machine learning. *Computational and structural biotechnology journal*, *18*, 3287-3300.

Appl, C., Moser, A., Baganz, F., & Hass, V. C. (2021). Digital Twins for bioprocess control strategy development and realisation. *Digital Twins: Applications to the Design and Optimization of Bioprocesses*, 63-94.

Ariño, J., Ramos, J., & Sychrová, H. (2010). Alkali metal cation transport and homeostasis in yeasts. *Microbiology and Molecular Biology Reviews*, *74*(1), 95-120.

Ataai, M. M., & Shuler, M. L. (1985). Simulation of CFSTR through development of a mathematical model for anaerobic growth of Escherichia coli cell population. *Biotechnology and bioengineering*, *27*(7), 1051-1055.

Bäck, T., Fogel, D. B., & Michalewicz, Z. (2000). Introduction to evolutionary algorithms. *Evolutionary computation*, *1*, 59-63.

Badr, S., & Sugiyama, H. (2020). A PSE perspective for the efficient production of monoclonal antibodies: integration of process, cell, and product design aspects. *Current Opinion in Chemical Engineering*, *27*, 121-128.

Badr, S., Okamura, K., Takahashi, N., Ubbenjans, V., Shirahata, H., & Sugiyama, H. (2021). Integrated design of biopharmaceutical manufacturing processes: Operation modes and process configurations for monoclonal antibody production. *Computers & Chemical Engineering*, *153*, 107422.

Bailey, J. E., & Ollis, D. F. (1986). *Biochemical engineering fundamentals*. McGraw-Hill.

Baker, R. E., Pena, J. M., Jayamohan, J., & Jérusalem, A. (2018). Mechanistic models versus machine learning, a fight worth fighting for the biological community?. *Biology letters*, *14*(5), 20170660.

Banga, J. R., Balsa-Canto, E., Moles, C. G., & Alonso, A. A. (2003). Dynamic optimization of bioreactors: a review. *Proceedings-Indian national science academy part A*, *69*(3/4), 257-266.

Banga, J. R., Balsa-Canto, E., Moles, C. G., & Alonso, A. A. (2005). Dynamic optimization of bioprocesses: Efficient and robust numerical strategies. *Journal of biotechnology*, *117*(4), 407-419.

Bangi, M. S. F., & Kwon, J. S. I. (2020). Deep hybrid modeling of chemical process: Application to hydraulic fracturing. *Computers & Chemical Engineering*, *134*, 106696.

Bangi, M. S. F., & Kwon, J. S. I. (2023). Deep hybrid model-based predictive control with guarantees on domain of applicability. *AIChE Journal*, *69*(5), e18012.

Banks, H. T., & Joyner, M. L. (2017). AIC under the framework of least squares estimation. *Applied Mathematics Letters*, *74*, 33-45.

Bapat, P. M., & Wangikar, P. P. (2004). Optimization of rifamycin B fermentation in shake flasks via a machine-learning-based approach. *Biotechnology and bioengineering*, *86*(2), 201-208.

Bastin, G. (2013). *On-line estimation and adaptive control of bioreactors* (Vol. 1). Elsevier.

Baughman, D. R., & Liu, Y. A. (1994). An expert network for predictive modeling and optimal design of extractive bioseparations in aqueous two-phase systems. *Industrial & engineering chemistry research*, *33*(11), 2668-2687.

Bayer, B., Dalmau Diaz, R., Melcher, M., Striedner, G., & Duerkop, M. (2021). Digital twin application for model-based doe to rapidly identify ideal process conditions for space-time yield optimization. *Processes*, *9*(7), 1109.

Bayer, B., Duerkop, M., Pörtner, R., & Möller, J. (2023). Comparison of mechanistic and hybrid modeling approaches for characterization of a CHO cultivation process: Requirements, pitfalls and solution paths. *Biotechnology journal*, *18*(1), 2200381.

Bayer, B., Duerkop, M., Striedner, G., & Sissolak, B. (2021). Model transferability and reduced experimental burden in cell culture process development facilitated by hybrid modeling and intensified design of experiments. *Frontiers in Bioengineering and Biotechnology*, *9*, 740215.

Bejani, M. M., & Ghatee, M. (2021). A systematic review on overfitting control in shallow and deep neural networks. *Artificial Intelligence Review*, 1-48.

Bornstein, B. J., Keating, S. M., Jouraku, A., & Hucka, M. (2008). LibSBML: an API library for SBML. *Bioinformatics*, *24*(6), 880-881.

Brady, C. P., Shimp, R. L., Miles, A. P., Whitmore, M., & Stowers, A. W. (2001). High-level production and purification of P30P2MSP119, an important vaccine antigen for malaria, expressed in the methylotropic yeast Pichia pastoris. *Protein expression and purification*, *23*(3), 468-475.

Breiman, L. (2001). Random forests. *Machine learning*, *45*, 5-32.

Breve, F. A., & Pedronette, D. C. G. (2016, September). Combined unsupervised and semi-supervised learning for data classification. In *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)* (pp. 1-6). Ieee.

Brunner, V., Siegl, M., Geier, D., & Becker, T. (2020). Biomass soft sensor for a Pichia pastoris fed-batch process based on phase detection and hybrid modeling. *Biotechnology and bioengineering*, *117*(9), 2749-2759.

Buck, K. K., Subramanian, V., & Block, D. E. (2002). Identification of critical batch operating parameters in fed-batch recombinant E. coli fermentations using decision tree analysis. *Biotechnology progress*, *18*(6), 1366-1376.

Butcher, B. and Smith, B.J. (2020) Feature Engineering and Selection: A Practical Approach for Predictive Models. American Statistician 74(3), 308-309.

Carinhas, N., Bernal, V., Teixeira, A. P., Carrondo, M. J., Alves, P. M., & Oliveira, R. (2011). Hybrid metabolic flux analysis: combining stoichiometric and statistical constraints to model the formation of complex recombinant products. *BMC systems biology*, *5*(1), 1-13.

Carrondo, M. J., Alves, P. M., Carinhas, N., Glassey, J., Hesse, F., Merten, O. W., ... & Mandenius, C. F. (2012). How can measurement, monitoring, modeling and control advance cell culture in industrial biotechnology?. *Biotechnology Journal*, *7*(12), 1522-1529.

Cereghino, G. P. L., Cereghino, J. L., Ilgen, C., & Cregg, J. M. (2002). Production of recombinant proteins in fermenter cultures of the yeast Pichia pastoris. *Current opinion in biotechnology*, *13*(4), 329-332.

Cereghino, J. L., & Cregg, J. M. (2000). Heterologous protein expression in the methylotrophic yeast Pichia pastoris. *FEMS microbiology reviews*, *24*(1), 45-66.

CHASSAGNOLE, C., FELL, D. A., RAÏS, B., KUDLA, B., & MAZAT, J. P. (2001). Control of the threonine-synthesis pathway in Escherichia coli: a theoretical and experimental approach. *Biochemical Journal*, *356*(2), 433-444.

Chen, L., Bernard, O., Bastin, G., & Angelov, P. (2000). Hybrid modelling of biotechnological processes using neural networks. *Control Engineering Practice*, *8*(7), 821-827.

Chen, L., Bernard, O., Bastin, G., & Angelov, P. (2000). Hybrid modelling of biotechnological processes using neural networks. *Control Engineering Practice*, *8*(7), 821-827.

Chen, L., Hontoir, Y., Huang, D., Zhang, J., & Morris, A. J. (2004). Combining first principles with black-box techniques for reaction systems. *Control Engineering Practice*, *12*(7), 819-826.

Clark, D. S., & Blanch, H. W. (1996). *Biochemical engineering*. CRC press.

Cook, C. E., Bergman, M. T., Cochrane, G., Apweiler, R., & Birney, E. (2018). The European Bioinformatics Institute in 2017: data coordination and integration. *Nucleic acids research*, *46*(D1), D21-D29.

Cornish-Bowden, A. (1995). Metabolic control analysis in theory and practice. In *Advances in molecular and cell biology* (Vol. 11, pp. 21-64). Elsevier.

Cos, O., Ramón, R., Montesinos, J. L., & Valero, F. (2006). Operational strategies, monitoring and control of heterologous protein production in the methylotrophic yeast Pichia pastoris under different promoters: a review. *Microbial cell factories*, *5*(1), 1-20.

Coşgun, A., Günay, M. E., & Yıldırım, R. (2021). Exploring the critical factors of algal biomass and lipid production for renewable fuel production by machine learning. *Renewable Energy*, *163*, 1299-1317.

Costa, R. S., Machado, D., Rocha, I., & Ferreira, E. C. (2010). Hybrid dynamic modeling of Escherichia coli central metabolic network combining Michaelis–Menten and approximate kinetic equations. *Biosystems*, *100*(2), 150-157.

Cuperlovic-Culf, M., Nguyen-Tran, T., & Bennett, S. A. (2022). Machine learning and hybrid methods for metabolic pathway modeling. *Computational Biology and Machine Learning for Metabolic Engineering and Synthetic Biology*, 417-439.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, *2*(4), 303-314.

Damasceno, L. M., Pla, I., Chang, H. J., Cohen, L., Ritter, G., Old, L. J., & Batt, C. A. (2004). An optimized fermentation process for high-level production of a single-chain Fv antibody fragment in Pichia pastoris. *Protein expression and purification*, *37*(1), 18-26.

Danø, S., Madsen, M. F., Schmidt, H., & Cedersund, G. (2006). Reduction of a biochemical model with preservation of its basic dynamic properties. *The FEBS journal*, *273*(21), 4862-4877.

De Brabander, P., Uitterhaegen, E., Delmulle, T., De Winter, K., & Soetaert, W. (2023). Challenges and progress towards industrial recombinant protein production in yeasts: A review. *Biotechnology Advances*, 108121.

Delalleau, O., & Bengio, Y. (2011). Shallow vs. deep sum-product networks. *Advances in neural information processing systems*, *24*.

DeLisle, R. K., & Dixon, S. L. (2004). Induction of decision trees via evolutionary programming. *Journal of chemical information and computer sciences*, *44*(3), 862-870.

Delvigne, F., Zune, Q., Lara, A. R., Al-Soud, W., & Sørensen, S. J. (2014). Metabolic variability in bioprocessing: implications of microbial phenotypic heterogeneity. *Trends in Biotechnology*, *32*(12), 608-616.

Depue, R. H., & Moat, A. G. (1961). Factors affecting aspartase activity. *Journal of Bacteriology*, *82*(3), 383-386.

Di Massimo, C., Montague, G. A., Willis, M. J., Tham, M. T., & Morris, A. J. (1992). Towards improved penicillin fermentation via artificial neural networks. *Computers & chemical engineering*, *16*(4), 283-291.

Dochain, D., Babary, J. P., & Tali-Maamar, N. (1992). Modelling and adaptive control of nonlinear distributed parameter bioreactors via orthogonal collocation. *Automatica*, *28*(5), 873-883.

Domach, M. M., & Shuler, M. L. (1984). A finite representation model for an asynchronous culture of E. coli. *Biotechnology and bioengineering*, *26*(8), 877-884.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE transactions on systems, man, and cybernetics, part b (cybernetics)*, *26*(1), 29-41.

Doyle, K., Tsopanoglou, A., Fejér, A., Glennon, B., & del Val, I. J. (2023). Automated assembly of hybrid dynamic models for CHO cell culture processes. *Biochemical Engineering Journal*, *191*, 108763.

efzi, H., Ang, K. S., Hanscho, M., Bordbar, A., Ruckerbauer, D., Lakshmanan, M., ... & Lewis, N. E. (2016). A consensus genome-scale reconstruction of Chinese hamster ovary cell metabolism. *Cell systems*, *3*(5), 434-443.

Eldan, R., & Shamir, O. (2016, June). The power of depth for feedforward neural networks. In *Conference on learning theory* (pp. 907-940). PMLR.

Fadda, S., Cincotti, A., & Cao, G. (2012). A novel population balance model to investigate the kinetics of in vitro cell proliferation: Part I. model development. *Biotechnology and Bioengineering*, *109*(3), 772-781.

Ferreira, A. R., Ataíde, F., Von Stosch, M., Dias, J. M. L., Clemente, J. J., Cunha, A. E., & Oliveira, R. (2012). Application of adaptive DO-stat feeding control to Pichia pastoris X33 cultures expressing a single chain antibody fragment (scFv). *Bioprocess and biosystems engineering*, *35*, 1603-1614.

Ferreira, A. R., Dias, J. M. L., Von Stosch, M., Clemente, J., Cunha, A. E., & Oliveira, R. (2014). Fast development of Pichia pastoris GS115 Mut+ cultures employing batch-to-batch control and hybrid semi-parametric modeling. *Bioprocess and biosystems engineering*, *37*, 629-639.

Fiedler, B., & Schuppert, A. (2008). Local identification of scalar hybrid models with tree structure. *IMA Journal of Applied Mathematics*, *73*(3), 449-476.

Folch-Fortuny, A., Marques, R., Isidro, I. A., Oliveira, R., & Ferrer, A. (2016). Principal elementary mode analysis (PEMA). *Molecular BioSystems*, *12*(3), 737-746.

Förster, J., Famili, I., Fu, P., Palsson, B. Ø., & Nielsen, J. (2003). Genome-scale reconstruction of the Saccharomyces cerevisiae metabolic network. *Genome research*, *13*(2), 244-253.

Galleguillos, S. N., Ruckerbauer, D., Gerstl, M. P., Borth, N., Hanscho, M., & Zanghellini, J. (2017). What can mathematical modelling say about CHO metabolism and protein glycosylation?. *Computational and structural biotechnology journal*, *15*, 212-221

Galvanauskas, V., Simutis, R., & Lübbert, A. (2004). Hybrid process models for process optimisation, monitoring and control. *Bioprocess and Biosystems Engineering*, *26*, 393-400.

Ganusov, V. V., Bril'kov, A. V., & Pechurkin, N. S. (2000). Mathematical modeling of population dynamics of unstable plasmid-containing bacteria during continuous cultivation in a chemostat. *Biofizika*, *45*(5), 908-914.

Ghosalkar, A., Sahai, V., & Srivastava, A. (2008). Optimization of chemically defined medium for recombinant Pichia pastoris for biomass production. *Bioresource technology*, *99*(16), 7906-7910.

Glassey, J., & Von Stosch, M. (Eds.). (2018). *Hybrid modeling in process industries*. CRC Press.

Glorot, X., & Bengio, Y. (2010, March). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics* (pp. 249-256). JMLR Workshop and Conference Proceedings.

Gnoth, S., Jenzsch, M., Simutis, R., & Lübbert, A. (2008). Product formation kinetics in genetically modified E. coli bacteria: inclusion body formation. *Bioprocess and biosystems engineering*, *31*, 41-46.

Gnoth, S., Simutis, R., & Lübbert, A. (2010). Selective expression of the soluble product fraction in Escherichia coli cultures employed in recombinant protein production processes. *Applied microbiology and biotechnology*, *87*, 2047-2058.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Goodman, A. G., Tanner, B. C., Chang, S. T., Esteban, M., & Katze, M. G. (2011). Virus infection rapidly activates the P58IPK pathway, delaying peak kinase activation to enhance viral replication. *Virology*, *417*(1), 27-36.

Greener, J. G., Kandathil, S. M., Moffat, L., & Jones, D. T. (2022). A guide to machine learning for biologists. *Nature Reviews Molecular Cell Biology*, *23*(1), 40-55.

Han, K., & Levenspiel, O. (1988). Extended Monod kinetics for substrate, product, and cell inhibition. *Biotechnology and bioengineering*, *32*(4), 430-447.

Hartmann, F. S., Udugama, I. A., Seibold, G. M., Sugiyama, H., & Gernaey, K. V. (2022). Digital models in biotechnology: Towards multi-scale integration and implementation. *Biotechnology Advances*, 108015.

Haykin, S. (2009). *Neural networks and learning machines, 3/E*. Pearson Education India.

Helleckes, L. M., Hemmerich, J., Wiechert, W., von Lieres, E., & Grünberger, A. (2023). Machine learning in bioprocess development: From promise to practice. *Trends in Biotechnology*, *41*(6), 817-835.

Henson, M. A. (2003). Dynamic modeling and control of yeast cell populations in continuous biochemical reactors. *Computers & Chemical Engineering*, *27*(8-9), 1185-1199.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

Hiscock, T. W. (2019). Adapting machine-learning algorithms to design gene circuits. *BMC bioinformatics*, *20*, 1-13.

Hjortso, M. A., & Nielsen, J. (1995). Population balance models of autonomous microbial oscillations. *Journal of biotechnology*, *42*(3), 255-269.

Hole, G., Hole, A. S., & McFalone-Shaw, I. (2021). Digitalization in pharmaceutical industry: What to focus on under the digital implementation process?. *International Journal of Pharmaceutics: X*, *3*, 100095.

Holland, J. H. (1992). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., ... & Kummer, U. (2006). COPASI—a complex pathway simulator. *Bioinformatics*, *22*(24), 3067-3074.

Hoskins, J. C., & Himmelblau, D. M. (1992). Process control via artificial neural networks and reinforcement learning. *Computers & chemical engineering*, *16*(4), 241-251.

Hucka, M., Finney, A., Sauro, H. M., Bolouri, H., Doyle, J. C., Kitano, H., ... & Wang, J. (2003). The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, *19*(4), 524-531.

Hutter, C., von Stosch, M., Cruz Bournazou, M. N., & Butté, A. (2021). Knowledge transfer across cell lines using hybrid Gaussian process models with entity embedding vectors. *Biotechnology and Bioengineering*, *118*(11), 4389-4401.

Hwang, T. M., Oh, H., Choi, Y. J., Nam, S. H., Lee, S., & Choung, Y. K. (2009). Development of a statistical and mathematical hybrid model to predict membrane fouling and performance. *Desalination*, *247*(1-3), 210-221.

Isidro, I. A., Portela, R. M., Clemente, J. J., Cunha, A. E., & Oliveira, R. (2016). Hybrid metabolic flux analysis and recombinant protein prediction in Pichia pastoris X-33 cultures expressing a single-chain antibody fragment. *Bioprocess and biosystems engineering*, *39*, 1351-1363.

Jahic, M., Knoblechner, J., Charoenrat, T., Enfors, S. O., & Veide, A. (2006). Interfacing Pichia pastoris cultivation with expanded bed adsorption. *Biotechnology and bioengineering*, *93*(6), 1040-1049.

Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. *Science*, *349*(6245), 255-260.

Joseph, B., & Hanratty, F. W. (1993). Predictive control of quality in a batch manufacturing process using artificial neural network models. *Industrial & engineering chemistry research*, *32*(9), 1951-1961.

Jović, A., Brkić, K., & Bogunović, N. (2015, May). A review of feature selection methods with applications. In *2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO)* (pp. 1200-1205). Ieee.

Kahrs, O., & Marquardt, W. (2008). Incremental identification of hybrid process models. *Computers & Chemical Engineering*, *32*(4-5), 694-705.

Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks* (Vol. 4, pp. 1942-1948). IEEE.

Kim, Y., Kim, G. B., & Lee, S. Y. (2021). Machine learning applications in genome-scale metabolic modeling. *Current Opinion in Systems Biology*, *25*, 42-49.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, *220*(4598), 671-680.

Klimasauskas, C. C. (1998). Hybrid modeling for robust nonlinear multivariable control. *ISA transactions*, *37*(4), 291-297.

König, M., Dräger, A., & Holzhütter, H. G. (2012). CySBML: a Cytoscape plugin for SBML. *Bioinformatics*, *28*(18), 2402-2403.

Kotidis, P., Jedrzejewski, P., Sou, S. N., Sellick, C., Polizzi, K., Del Val, I. J., & Kontoravdi, C. (2019). Model-based optimization of antibody galactosylation in CHO cell culture. *Biotechnology and bioengineering*, *116*(7), 1612-1626.

Kotidis, P., Pappas, I., Avraamidou, S., Pistikopoulos, E. N., Kontoravdi, C., & Papathanasiou, M. M. (2021). DigiGlyc: A hybrid tool for reactive scheduling in cell culture systems. *Computers & Chemical Engineering*, *154*, 107460.

Koutsoukas, A., Monaghan, K. J., Li, X., & Huan, J. (2017). Deep-learning: investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data. *Journal of cheminformatics*, *9*(1), 1-13.

Krogh, A. (2008). What are artificial neural networks?. *Nature biotechnology*, *26*(2), 195-197.

Kurz, S., De Gersem, H., Galetzka, A., Klaedtke, A., Liebsch, M., Loukrezis, D., ... & Schmidt, M. (2022). Hybrid modeling: towards the next level of scientific computing in engineering. *Journal of Mathematics in Industry*, *12*(1), 1-12.

Larochelle, H., Bengio, Y., Louradour, J., & Lamblin, P. (2009). Exploring strategies for training deep neural networks. *Journal of machine learning research*, *10*(1).

Larranaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., ... & Robles, V. (2006). Machine learning in bioinformatics. *Briefings in bioinformatics*, *7*(1), 86-112.

Lawrence, S., Giles, C. L., & Tsoi, A. C. (1997, July). Lessons in neural network training: Overfitting may be harder than expected. In *Aaai/iaai* (pp. 540-545).

Lawrence, S., Giles, C. L., & Tsoi, A. C. (1998). *What size neural network gives optimal generalization? Convergence properties of backpropagation*.

Le Novere, N., Bornstein, B., Broicher, A., Courtot, M., Donizelli, M., Dharuri, H., ... & Hucka, M. (2006). BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic acids research*, *34*(suppl_1), D689-D691.

Le, N. Q. K. (2022). Potential of deep representative learning features to interpret the sequence information in proteomics. *Proteomics*, *22*(1-2), 2100232.

Le, N. Q. K., Do, D. T., Hung, T. N. K., Lam, L. H. T., Huynh, T. T., & Nguyen, N. T. K. (2020). A computational framework based on ensemble deep neural networks for essential genes identification. *International journal of molecular sciences*, *21*(23), 9070.

Lee, D. S., Vanrolleghem, P. A., & Park, J. M. (2005). Parallel hybrid modeling methods for a full-scale cokes wastewater treatment plant. *Journal of Biotechnology*, *115*(3), 317-328.

Lee, D., Jayaraman, A., & Kwon, J. S. (2020). Development of a hybrid model for a partially known intracellular signaling pathway through correction term estimation and neural network modeling. *PLoS Computational Biology*, *16*(12), e1008472.

Lee, J. H., Shin, J., & Realff, M. J. (2018). Machine learning: Overview of the recent progresses and implications for the process systems engineering field. *Computers & Chemical Engineering*, *114*, 111-121.

Lee, J., & Ramirez, W. F. (1994). Optimal fed-batch control of induced foreign protein production by recombinant bacteria. *AIChE Journal*, *40*(5), 899-907.

Lewis, J. E., & Kemp, M. L. (2021). Integration of machine learning and genome-scale metabolic modeling identifies multi-omics biomarkers for radiation resistance. *Nature Communications*, *12*(1), 2700.

Li, B., Morris, J., & Martin, E. B. (2002). Model selection for partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, *64*(1), 79-89.

Liang, S., & Srikant, R. (2016). Why deep neural networks for function approximation?. *arXiv preprint arXiv:1610.04161*.

Liebermeister, W., & Klipp, E. (2006). Bringing metabolic networks to life: convenience rate law and thermodynamic constraints. *Theoretical Biology and Medical Modelling*, *3*, 1-13.

Liebermeister, W., Uhlendorf, J., & Klipp, E. (2010). Modular rate laws for enzymatic reactions: thermodynamics, elasticities and implementation. *Bioinformatics*, *26*(12), 1528-1534.

Liu, W. K., Li, S., & Belytschko, T. (1997). Moving least-square reproducing kernel methods (I) methodology and convergence. *Computer methods in applied mechanics and engineering*, *143*(1-2), 113-154.

Luedeking, R., & Piret, E. L. (1959). A kinetic study of the lactic acid fermentation. Batch process at controlled pH. *Journal of Biochemical and Microbiological Technology and Engineering*, *1*(4), 393-412.

Lukowski, G., Rauch, A., & Rosendahl, T. (2019). The virtual representation of the world is emerging. *Future Telco: Successful Positioning of Network Operators in the Digital Age*, 165-173

Luo, Z., Liu, H., & Wu, X. (2005, July). Artificial neural network computation on graphic process unit. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.* (Vol. 1, pp. 622-626). IEEE.

Luus, R. (1992). On the application of iterative dynamic programming to singular optimal control problems. *IEEE transactions on automatic control*, *37*(11), 1802-1806.

Ma, Y., Noreña-Caro, D. A., Adams, A. J., Brentzel, T. B., Romagnoli, J. A., & Benton, M. G. (2020). Machine-learning-based simulation and fed-batch control of cyanobacterial-phycocyanin production in Plectonema by artificial neural network and deep reinforcement learning. *Computers & Chemical Engineering*, *142*, 107016.

Mahadevan, R., Edwards, J. S., & Doyle, F. J. (2002). Dynamic flux balance analysis of diauxic growth in Escherichia coli. *Biophysical journal*, *83*(3), 1331-1340.

Mahalec, V., & Sanchez, Y. (2012). Inferential monitoring and optimization of crude separation units via hybrid models. *Computers & chemical engineering*, *45*, 15-26.

Mandenius, C. F. (2004). Recent developments in the monitoring, modeling and control of biological production systems. *Bioprocess and Biosystems Engineering*, *26*, 347-351.

Mantzaris, N. V., Daoutidis, P., & Srienc, F. (2001). Numerical solution of multi-variable cell population balance models: I. Finite difference methods. *Computers & Chemical Engineering*, *25*(11-12), 1411-1440.

Martínez-Muñoz, G. A., & Pena, A. (2005). In situ study of K+ transport into the vacuole of Saccharomyces cerevisiae. *Yeast*, *22*(9), 689-704.

Masri, S. F. (1994). A hybrid parametric/nonparametric approach for the identification of non-linear systems. *Probabilistic Engineering Mechanics*, *9*(1-2), 47-57.

McLamore, E. S., Huffaker, R., Shupler, M., Ward, K., Datta, S. P. A., Katherine Banks, M., ... & Foster, J. S. (2020). Digital Proxy of a Bio-Reactor (DIYBOT) combines sensor data and data analytics to improve greywater treatment and wastewater management systems. *Scientific Reports*, *10*(1), 8015.

Mei, X., & Smith, P. K. (2021). A comparison of in-sample and out-of-sample model selection approaches for artificial neural network (ANN) daily streamflow simulation. *Water*, *13*(18), 2525.

Melcher, M., Scharl, T., Spangl, B., Luchner, M., Cserjan, M., Bayer, K., ... & Striedner, G. (2015). The potential of random forest and neural networks for biomass and recombinant protein modeling in Escherichia coli fed-batch fermentations. *Biotechnology Journal*, *10*(11), 1770-1782.

Mhaskar, H. N., & Poggio, T. (2016). Deep vs. shallow networks: An approximation theory perspective. *Analysis and Applications*, *14*(06), 829-848.

Mochão, H., Barahona, P., & Costa, R. S. (2020). KiMoSys 2.0: an upgraded database for submitting, storing and accessing experimental data for kinetic modeling. *Database*, *2020*, baaa093.

Monk, J. M., Charusanti, P., Aziz, R. K., Lerman, J. A., Premyodhin, N., Orth, J. D., ... & Palsson, B. Ø. (2013). Genome-scale metabolic reconstructions of multiple Escherichia coli strains highlight strain-specific adaptations to nutritional environments. *Proceedings of the National Academy of Sciences*, *110*(50), 20338-20343.

Monod, J. (1949). The growth of bacterial cultures: Annual Reviews in Microbiology, v. 3.

Monteiro, M., Fadda, S., & Kontoravdi, C. (2023). Towards advanced bioprocess optimization: A multiscale modelling approach. *Computational and Structural Biotechnology Journal*, *21*, 3639-3655.

Mora, A., Zhang, S., Carson, G., Nabiswa, B., Hossler, P., & Yoon, S. (2018). Sustaining an efficient and effective CHO cell line development platform by incorporation of 24-deep well plate screening and multivariate analysis. *Biotechnology Progress*, *34*(1), 175-186.

Moser, A., Appl, C., Brüning, S., & Hass, V. C. (2021). Mechanistic mathematical models as a basis for digital twins. *Digital Twins: Tools and Concepts for Smart Biomanufacturing*, 133-180.

Mowbray, M., Vallerio, M., Perez-Galvan, C., Zhang, D., Chanona, A. D. R., & Navarro-Brull, F. J. (2022). Industrial data science–a review of machine learning applications for chemical and process industries. *Reaction Chemistry & Engineering*, *7*(7), 1471-1509.

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)* (pp. 807-814).

Narayanan, H., Luna, M., Sokolov, M., Butte, A., & Morbidelli, M. (2022). Hybrid models based on machine learning and an increasing degree of process knowledge: application to cell culture processes. *Industrial & Engineering Chemistry Research*, *61*(25), 8658-8672.

Narayanan, H., Sokolov, M., Morbidelli, M., & Butté, A. (2019). A new generation of predictive models: The added value of hybrid models for manufacturing processes of therapeutic proteins. *Biotechnology and bioengineering*, *116*(10), 2540-2549.

Narayanan, H., von Stosch, M., Feidl, F., Sokolov, M., Morbidelli, M., & Butté, A. (2023). Hybrid modeling for biopharmaceutical processes: advantages, opportunities, and implementation. *Frontiers in Chemical Engineering*, *5*, 1157889.

Narendra, K. S. (1989). Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, *1*(1), 183-192.

Nargund, S., Guenther, K., & Mauch, K. (2019). The move toward biopharma 4.0: insilico biotechnology develops "smart" processes that benefit biomanufacturing through digital twins. *Genetic Engineering & Biotechnology News*, *39*(6), 53-55.

Nascimento, C. A. O., Giudici, R., & Scherbakoff, N. (1999). Modeling of industrial nylon-6, 6 polymerization process in a twin-screw extruder reactor. II. Neural networks and hybrid models. *Journal of applied polymer science*, *72*(7), 905-912.

Nian, R., Liu, J., & Huang, B. (2020). A review on reinforcement learning: Introduction and applications in industrial process control. *Computers & Chemical Engineering*, *139*, 106886.

Nielsen, J., Villadsen, J., Nielsen, J., & Villadsen, J. (1994). Population Balances Based on Cell Number. *Bioreaction Engineering Principles*, 271-294.

Nishimura, Y., & Bailey, J. E. (1981). Bacterial population dynamics in batch and continuous-flow microbial reactors. *AIChE Journal*, *27*(1), 73-81.

Nocedal, J., & Wright, S. J. (Eds.). (1999). *Numerical optimization*. New York, NY: Springer New York.

Nold, V., Junghans, L., Bayer, B., Bisgen, L., Duerkop, M., Drerup, R., ... & Knapp, B. (2023). Boost dynamic protocols for producing mammalian biopharmaceuticals with intensified

DoE—a practical guide to analyses with OLS and hybrid modeling. *Frontiers in Chemical Engineering*, *4*, 1044245.

O'Brien, C. M., Zhang, Q., Daoutidis, P., & Hu, W. S. (2021). A hybrid mechanistic-empirical model for in silico mammalian cell bioprocess simulation. *Metabolic Engineering*, *66*, 31-40.

Okamura, K., Badr, S., Murakami, S., & Sugiyama, H. (2022). Hybrid Modeling of CHO Cell Cultivation in Monoclonal Antibody Production with an Impurity Generation Module. *Industrial & Engineering Chemistry Research*, *61*(40), 14898-14909.

Okorokov, L. A., & Lehle, L. (1998). Ca2+-ATPases of Saccharomyces cerevisiae: diversity and possible role in protein sorting. *FEMS microbiology letters*, *162*(1), 83-91.

Oliveira, R. (2004). Combining first principles modelling and artificial neural networks: a general framework. *Computers & Chemical Engineering*, *28*(5), 755-766.

Olivier, B. G., & Snoep, J. L. (2004). Web-based kinetic modelling using JWS Online. *Bioinformatics*, *20*(13), 2143-2144.

Palsson, B. (2002). In silico biology through "omics". *Nature biotechnology*, *20*(7), 649-650.

Park, S., & Fred Ramirez, W. (1988). Optimal production of secreted protein in fed-batch reactors. *AIChE Journal*, *34*(9), 1550-1558.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, *12*, 2825-2830.

Peres, J., Oliveira, R., & de Azevedo, S. F. (2008). Bioprocess hybrid parametric/nonparametric modelling based on the concept of mixture of experts. *Biochemical Engineering Journal*, *39*(1), 190-206.

Pigou, M., Morchain, J., Fede, P., Penet, M. I., & Laronze, G. (2017). An assessment of methods of moments for the simulation of population dynamics in large-scale bioreactors. *Chemical Engineering Science*, *171*, 218-232.

Pinto, J., Antunes, J., Ramos, J., Costa, R. S., & Oliveira, R. (2022). Modeling and optimization of bioreactor processes. In *Current Developments in Biotechnology and Bioengineering* (pp. 89-115). Elsevier.

Pinto, J., Costa, R. S., Alexandre, L., Ramos, J., & Oliveira, R. (2023). SBML2HYB: a Python interface for SBML compatible hybrid modeling. *Bioinformatics*, *39*(1), btad044.

Pinto, J., de Azevedo, C. R., Oliveira, R., & von Stosch, M. (2019). A bootstrap-aggregated hybrid semi-parametric modeling framework for bioprocess development. *Bioprocess and biosystems engineering*, *42*, 1853-1865.

Pinto, J., Mestre, M., Ramos, J., Costa, R. S., Striedner, G., & Oliveira, R. (2022). A general deep hybrid model for bioreactor systems: Combining first principles with deep neural networks. *Computers & Chemical Engineering*, *165*, 107952.

Pinto, J., Ramos, J. R., Costa, R. S., & Oliveira, R. (2023). A General Hybrid Modeling Framework for Systems Biology Applications: Combining Mechanistic Knowledge with Deep Neural Networks under the SBML Standard. *AI*, *4*(1), 303-318. and

Pinto, J., Ramos, J. R., Costa, R. S., & Oliveira, R. (2023). Hybrid Deep Modeling of a GS115 (Mut+) Pichia pastoris Culture with State–Space Reduction. *Fermentation*, *9*(7), 643.

Pinto, J., Ramos, J. R., Costa, R. S., Rossell, S., Dumas, P., & Oliveira, R. (2023). Hybrid deep modeling of a CHO-K1 fed-batch process: combining first-principles with deep neural networks. *Frontiers in Bioengineering and Biotechnology*, *11*.

Plantz, B. A., Nickerson, K., Kachman, S. D., & Schlegel, V. L. (2007). Evaluation of Metals in a Defined Medium for Pichia pastoris Expressing Recombinant β-Galactosidase. *Biotechnology progress*, *23*(3), 687-692.

Potočnik, P., & Grabec, I. (1999). Empirical modeling of antibiotic fermentation process using neural networks and genetic algorithms. *Mathematics and computers in simulation*, *49*(4-5), 363-379.

Powell, K. M., Machalek, D., & Quah, T. (2020). Real-time optimization using reinforcement learning. *Computers & Chemical Engineering*, *143*, 107077.

Preusting, H., Noordover, J., Simutis, R., & Lübbert, A. (1996). The use of hybrid modelling for the optimization of the penicillin fermentation process. *Chimia*, *50*(9), 416-416.

Psichogios, D. C., & Ungar, L. H. (1992). A hybrid neural network-first principles approach to process modeling. *AIChE Journal*, *38*(10), 1499-1511.

Qin, Y., Yalamanchili, H. K., Qin, J., Yan, B., & Wang, J. (2015). The current status and challenges in computational analysis of genomic big data. *Big data research*, *2*(1), 12-18.

Quek, L. E., Dietmair, S., Hanscho, M., Martínez, V. S., Borth, N., & Nielsen, L. K. (2014). Reducing Recon 2 for steady-state flux analysis of HEK cell culture. *Journal of biotechnology*, *184*, 172-178.

Quiza, R., López-Armas, O., & Davim, J. P. (2012). *Hybrid modeling and optimization of manufacturing: Combining artificial intelligence and finite element method*. Springer Science & Business Media.

Rajulapati, L., Chinta, S., Shyamala, B., & Rengaswamy, R. (2022). Integration of machine learning and first principles models. *AIChE Journal*, *68*(6), e17715.

Ramos, J. R., Oliveira, G. P., Dumas, P., & Oliveira, R. (2022). Genome-scale modeling of Chinese hamster ovary cells by hybrid semi-parametric flux balance analysis. *Bioprocess and biosystems engineering*, *45*(11), 1889-1904.

Rindskopf, D. (1997) An introduction to the bootstrap - Efron,B, Tibshirani,RJ. Journal of Educational and Behavioral Statistics 22(2), 245-245.

Robitaille, J., Chen, J., & Jolicoeur, M. (2015). A single dynamic metabolic model can describe mAb producing CHO cell batch and fed-batch cultures on different culture media. *PloS one*, *10*(9), e0136815.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533-536.

Salah, L. B., & Fourati, F. (2019, March). Deep MLP neural network control of bioreactor. In *2019 10th International Renewable Energy Congress (IREC)* (pp. 1-5). IEEE.

Sansana, J., Joswiak, M. N., Castillo, I., Wang, Z., Rendall, R., Chiang, L. H., & Reis, M. S. (2021). Recent trends on hybrid modeling for Industry 4.0. *Computers & Chemical Engineering*, *151*, 107365.

Savageau, M. A. (1970). Biochemical systems analysis: III. Dynamic solutions using a power-law approximation. *Journal of theoretical biology*, *26*(2), 215-226.

Schenker, B., & Agarwal, M. (2000). Online-optimized feed switching in semi-batch reactors using semi-empirical dynamic models. *Control Engineering Practice*, *8*(12), 1393-1403.

Schubert, J., Simutis, R., Dors, M., Havlik, I., & Lübbert, A. (1994). Bioprocess optimization and control: Application of hybrid modelling. *Journal of biotechnology*, *35*(1), 51-68.

Schubert, J., Simutis, R., Dors, M., Havlík, I., & Lübbert, A. (1994). Hybrid modelling of yeast production processes–combination of a priori knowledge on different levels of sophistication. *Chemical Engineering & Technology: Industrial Chemistry-Plant Equipment-Process Engineering-Biotechnology*, *17*(1), 10-20.

Schubert, J., Simutis, R., Dors, M., Havlík, I., & Lübbert, A. (1994). Hybrid modelling of yeast production processes–combination of a priori knowledge on different levels of sophistication. *Chemical Engineering & Technology: Industrial Chemistry-Plant Equipment-Process Engineering-Biotechnology*, *17*(1), 10-20.

Senger, R. S., & Karim, M. N. (2003). Neural-Network-Based Identification of Tissue-Type Plasminogen Activator Protein Production and Glycosylation in CHO Cell Culture under Shear Environment. *Biotechnology progress*, *19*(6), 1828-1836.

Seo, K. H., & Rhee, J. I. (2004). High-level expression of recombinant phospholipase C from Bacillus cereus in Pichia pastoris and its characterization. *Biotechnology letters*, *26*(19), 1475-1479.

Shin, J., Badgwell, T. A., Liu, K. H., & Lee, J. H. (2019). Reinforcement learning–overview of recent progress and implications for process control. *Computers & Chemical Engineering*, *127*, 282-294.

Shipman, J. W. (2010). *Tkinter reference: a GUI for Python, New Mexico Tech Computer Center* (pp. 12-12). retrieved 2012-01-11.

Shrivastava, R., Mahalingam, H., & Dutta, N. N. (2017). Application and evaluation of random forest classifier technique for fault detection in bioreactor operation. *Chemical Engineering Communications*, *204*(5), 591-598.

Sidoli, F. R., Mantalaris, A., & Asprey, S. (2004). Modelling of mammalian cells and cell culture processes. *Cytotechnology*, *44*(1-2), 27-46.

Singh, A., Thakur, N., & Sharma, A. (2016, March). A review of supervised machine learning algorithms. In *2016 3rd international conference on computing for sustainable global development (INDIACom)* (pp. 1310-1315). Ieee.

Singh, M., Singh, R., Singh, S., Walker, G., & Matsoukas, T. (2020). Discrete finite volume approach for multidimensional agglomeration population balance equation on unstructured grid. *Powder Technology*, *376*, 229-240.

Smiatek, J., Jung, A., & Bluhmki, E. (2020). Towards a digital bioprocess replica: computational approaches in biopharmaceutical development and manufacturing. *Trends in Biotechnology*, *38*(10), 1141-1153.

Sohn, S. B., Graf, A. B., Kim, T. Y., Gasser, B., Maurer, M., Ferrer, P., ... & Lee, S. Y. (2010). Genome-scale metabolic model of methylotrophic yeast Pichia pastoris and its use for in silico analysis of heterologous protein production. *Biotechnology journal*, *5*(7), 705-715.

Sokolov, M., von Stosch, M., Narayanan, H., Feidl, F., & Butté, A. (2021). Hybrid modeling—a key enabler towards realizing digital twins in biopharma?. *Current Opinion in Chemical Engineering*, *34*, 100715.

Spencer, J. F. (1997). *Yeasts in natural and artificial habitats*. Springer Science & Business Media.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, *15*(1), 1929-1958.

Stanford, N. J., Lubitz, T., Smallbone, K., Klipp, E., Mendes, P., & Liebermeister, W. (2013). Systematic construction of kinetic models from genome-scale metabolic networks. *PloS one*, *8*(11), e79195.

Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, *11*, 341-359.

Su, H. T., & McAvoy, T. J. (1993). Integration of multilayer perceptron networks and linear dynamic models: a Hammerstein modeling approach. *Industrial & engineering chemistry research*, *32*(9), 1927-1936.

Surribas, A., Stahn, R., Montesinos, J. L., Enfors, S. O., Valero, F., & Jahic, M. (2007). Production of a Rhizopus oryzae lipase from Pichia pastoris using alternative operational strategies. *Journal of biotechnology*, *130*(3), 291-299.

Sutton, R. S., & Barto, A. G. (1999). Reinforcement learning: An introduction. *Robotica*, *17*(2), 229-235.

Tashman, L. J. (2000). Out-of-sample tests of forecasting accuracy: an analysis and review. *International journal of forecasting*, *16*(4), 437-450.

Teixeira, A. P., Alves, C., Alves, P. M., Carrondo, M. J., & Oliveira, R. (2007). Hybrid elementary flux analysis/nonparametric modeling: application for bioprocess control. *BMC bioinformatics*, *8*(1), 1-15.

Teixeira, A. P., Carinhas, N., Dias, J. M., Cruz, P., Alves, P. M., Carrondo, M. J., & Oliveira, R. (2007). Hybrid semi-parametric mathematical systems: Bridging the gap between systems biology and process engineering. *Journal of biotechnology*, *132*(4), 418-425.

Teixeira, A. P., Clemente, J. J., Cunha, A. E., Carrondo, M. J., & Oliveira, R. (2006). Bioprocess iterative batch-to-batch optimization based on hybrid parametric/nonparametric models. *Biotechnology progress*, *22*(1), 247-258.

Teixeira, A. P., Dias, J. M., Carinhas, N., Sousa, M., Clemente, J. J., Cunha, A. E., ... & Oliveira, R. (2011). Cell functional enviromics: unravelling the function of environmental factors. *BMC systems biology*, *5*, 1-16.

Teixeira, A., Cunha, A. E., Clemente, J. J., Moreira, J. L., Cruz, H. J., Alves, P. M., ... & Oliveira, R. (2005). Modelling and optimization of a recombinant BHK-21 cultivation process using hybrid grey-box systems. *Journal of Biotechnology*, *118*(3), 290-303.

Teoh, E. J., Tan, K. C., & Xiang, C. (2006). Estimating the number of hidden neurons in a feedforward network using the singular value decomposition. *IEEE Transactions on Neural Networks*, *17*(6), 1623-1629.

Thiele, I., Swainston, N., Fleming, R. M., Hoppe, A., Sahoo, S., Aurich, M. K., ... & Palsson, B. Ø. (2013). A community-driven global reconstruction of human metabolism. *Nature biotechnology*, *31*(5), 419-425.

Tholudur, A., & Ramirez, W. F. (1996). Optimization of fed-batch bioreactors using neural network parameter function models. *Biotechnology Progress*, *12*(3), 302-309.

Thompson, M. L., & Kramer, M. A. (1994). Modeling chemical processes using prior knowledge and neural networks. *AIChE Journal*, *40*(8), 1328-1340.

Udaya Mohanan, K., Cho, S., & Park, B. G. (2023). Optimization of the structural complexity of artificial neural network for hardware-driven neuromorphic computing application. *Applied Intelligence*, *53*(6), 6288-6306.

Udugama, I. A., Öner, M., Lopez, P. C., Beenfeldt, C., Bayer, C., Huusom, J. K., ... & Sin, G. (2021). Towards digitalization in bio-manufacturing operations: a survey on application of big data and digital twin concepts in Denmark. *Front. Chem. Eng.*, *3*, 727152.

Umar, M., Sabir, Z., Raja, M. A. Z., Baskonus, H. M., Yao, S. W., & Ilhan, E. (2021). A novel study of Morlet neural networks to solve the nonlinear HIV infection system of latently infected cells. *Results in Physics*, *25*, 104235.

Umar, M., Sabir, Z., Raja, M. A. Z., Shoaib, M., Gupta, M., & Sánchez, Y. G. (2020). A stochastic intelligent computing with neuro-evolution heuristics for nonlinear SITR system of novel COVID-19 dynamics. *Symmetry*, *12*(10), 1628.

Van Can, H. J., Te Braake, H. A., Dubbelman, S., Hellinga, C., Luyben, K. C. A., & Heijnen, J. J. (1998). Understanding and applying the extrapolation properties of serial gray-box models. *AIChE journal*, *44*(5), 1071-1089.

Vcelar, S., Jadhav, V., Melcher, M., Auer, N., Hrdina, A., Sagmeister, R., ... & Borth, N. (2018). Karyotype variation of CHO host cell lines over time in culture characterized by chromosome counting and chromosome painting. *Biotechnology and bioengineering*, *115*(1), 165-173.

Vijayakumar, S., Rahman, P. K., & Angione, C. (2020). A hybrid flux balance analysis and machine learning pipeline elucidates metabolic adaptation in cyanobacteria. *Iscience*, *23*(12).

Visser, D., & Heijnen, J. J. (2003). Dynamic simulation and metabolic re-design of a branched pathway using linlog kinetics. *Metabolic engineering*, *5*(3), 164-176.

von Stosch, M., Hamelink, J. M., & Oliveira, R. (2016). Hybrid modeling as a QbD/PAT tool in process development: an industrial E. coli case study. *Bioprocess and biosystems engineering*, *39*, 773-784.

Von Stosch, M., Oliveira, R., Peres, J., & de Azevedo, S. F. (2014). Hybrid semi-parametric modeling in process systems engineering: Past, present and future. *Computers & Chemical Engineering*, *60*, 86-101.

von Stosch, M., Oliveira, R., Peres, J., & De Azevedo, S. F. (2011). A novel identification method for hybrid (N) PLS dynamical systems with application to bioprocesses. *Expert Systems with Applications*, *38*(9), 10862-10874.

Von Stosch, M., Peres, J., de Azevedo, S. F., & Oliveira, R. (2010). Modelling biochemical networks with intrinsic time delays: a hybrid semi-parametric approach. *BMC Systems Biology*, *4*(1), 1-13.

Wales, D. J., & Doye, J. P. (1997). Global optimization by basin-hopping and the lowest energy structures of Lennard-Jones clusters containing up to 110 atoms. *The Journal of Physical Chemistry A*, *101*(28), 5111-5116.

Walker, G. M., & Maynard, A. I. (1997). Accumulation of magnesium ions during fermentative metabolism in Saccharomyces cerevisiae. *Journal of Industrial Microbiology and Biotechnology*, *18*, 1-3.

Werbos, P. (1974). Beyond regression: New tools for prediction and analysis in the behavioral sciences. *PhD thesis, Committee on Applied Mathematics, Harvard University, Cambridge, MA*.

Willsky, G. R. (1979). Characterization of the plasma membrane Mg2+-ATPase from the yeast, Saccharomyces cerevisiae. *Journal of Biological Chemistry*, *254*(9), 3326-3332.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE transactions on evolutionary computation*, *1*(1), 67-82.

Wouwer, A. V., Renotte, C., & Bogaerts, P. (2004). Biological reaction modeling using radial basis function networks. *Computers & chemical engineering*, *28*(11), 2157-2164.

Wouwer, A. V., Renotte, C., Remy, M., & Bogaerts, P. (2001). Hybrid Physical-Neural Network Modeling of Animal Cell Cultures. *IFAC Proceedings Volumes*, *34*(22), 331-336.

Xiang, Y., Gubian, S., Suomela, B., & Hoeng, J. (2013). Generalized simulated annealing for global optimization: the GenSA package. *R J.*, *5*(1), 13.

Yang, C. T., Kristiani, E., Leong, Y. K., & Chang, J. S. (2023). Big Data and Machine Learning Driven Bioprocessing-Recent trends and critical analysis. *Bioresource technology*, 128625.

Yang, J. H., Wright, S. N., Hamblin, M., McCloskey, D., Alcantar, M. A., Schrübbers, L., ... & Collins, J. J. (2019). A white-box machine learning approach for revealing antibiotic mechanisms of action. *Cell*, *177*(6), 1649-1661.

Yang, O., Prabhu, S., & Ierapetritou, M. (2019). Comparison between batch and continuous monoclonal antibody production and economic analysis. *Industrial & Engineering Chemistry Research*, *58*(15), 5851-5863.

Yu, H., Wen, G., Gan, J., Zheng, W., & Lei, C. (2020). Self-paced learning for k-means clustering algorithm. *Pattern Recognition Letters*, *132*, 69-75.

Zhang, J., & Greasham, R. (1999). Chemically defined media for commercial fermentations. *Applied microbiology and biotechnology*, *51*, 407-421.

Zhang, L., Pan, M., Quan, S., Chen, Q., & Shi, Y. (2006, June). Adaptive neural control based on pemfc hybrid modeling. In *2006 6th World Congress on Intelligent Control and Automation* (Vol. 2, pp. 8319-8323). IEEE.

Zhao, H. L., Xue, C., Wang, Y., Yao, X. Q., & Liu, Z. M. (2008). Increasing the cell viability and heterologous protein expression of Pichia pastoris mutant deficient in PMR1 gene by culture condition optimization. *Applied microbiology and biotechnology*, *81*, 235-241.

Zhu, G. Y., Zamamiri, A., Henson, M. A., & Hjortsø, M. A. (2000). Model predictive control of continuous yeast bioreactors using cell population balance models. *Chemical Engineering Science*, *55*(24), 6155-6167.

Hybrid Deep Modeling of Biotechnological Processes: Combining Deep Neural Networks with First Principles

José Miguel Machado de Matos Pinto

2024