Applying Deep Learning for Streamlined Bioreactor Modelling and Control-Optimization

BREE 495 - Engineering Design 3
Department of Bioresource Engineering, McGill University, Macdonald Campus
Due April 16, 2022

Prof. Madramootoo
Amanda Crèche
Joel Harms

Mentor: Mohamed Debbagh

Abstract

Feedstock composition determines the control, output and efficiency of bioreactors; their characterization is therefore a crucial step in bioreactor operation. However, measurements of such characteristics may be costly. It may thus be economically challenging for a bioreactor-based company that is using various feedstock types to regularly conduct feedstock laboratory analysis. This project explores whether and how machine learning may aid with feedstock characterization, decrease its cost and make model predictive control more accessible. Three deep learning architectures were tested: a fully connected Multi-Layer-Perceptron model, a Convolutional Neural Network and a Recurrent Neural Network using Gated-Recurrent Units (GRUs). All models were able to achieve a high coefficient of determination for the feedstock parameters predicted, the mean R-square over all models and all variables predicted was 0.8961. The GRU model achieved the highest coefficient of determination on 4 out of 5 predicted feedstock parameters. It was concluded that deep learning is a promising tool that could assist in easing bioreaction optimisation. The models were deployed to a web-based graphical user interface to showcase the models and make the results of this research more accessible to the general and professional public. All code used to create the deep learning models and the interface were published through an open-source GitHub repository to serve as a template that may be used for applying this approach to commercial projects.

Table of Contents:

List of Tables

List of Figures

List of Acronyms and Abbreviations

ADAM: Adaptive moment estimation optimizer

ANN: Artificial neural network

API: Application programming interface

ASM: Activated Sludge Model

ASM1: Activated Sludge Model 1

C/N ratio: a carbon-to-nitrogen ratio

COD: Chemical oxygen demand

ConvNN: Convolutional neural network

DO: Dissolved oxygen

GPU: Graphics processing units

GRU: Gated recurrent units

GUI: Graphical user interface

H^+: hydrogen ion

MAE: Mean absolute error

MIT: Massachusetts Institute of Technology

ML: Machine Learning

MLP: Multi-layer-perceptron

MSE: Mean square error

N_2O: Nitrous oxide emissions

NH_4^+: Ammonium ion

NO_3^- : Nitrate ion

rbCOD: Readily biodegradable chemical oxygen demand

SBON: Soluble Biodegradable Organic Nitrogen

RNN: Recurrent neural network

UI: User Interface

Acknowledgements

We want to acknowledge the importance of the fastai Practical Deep Learning for Coders course taught by Jeremy Howard and the Applied Machine Learning class at McGill (COMP 551) taught by Prof. Yue Li, in allowing us to undertake this project. We would like to thank Professors Chandra A. Madramootoo, Mark Lefsrud, Shanpeng Sun, Grant Clark and Yue Li for taking the time to provide us with guidance and answer our questions. We, further, also would like to thank the team of Circulus AgTech for their time, and specifically Mohamed Debbagh, Vincent Desaulnier and David Leroux for working with us.

1. Introduction

1.1 Feedstock Parameters

Feedstock composition significantly affects the bioreaction process, its outputs and the operational energy it requires. In sewage treatment plants, factors such as the readily biodegradable COD, chemical oxygen demand in the water (rbCOD, “small molecules that are directly available for biodegradation by heterotrophic microorganisms” (Bolek, n.d.) can impede bioreactor efficiency. It has been determined that too high rbCOD and high food to microorganism ratio leads to system bulking and foaming. High rbCOD uptake uptake can also be responsible for creation of anoxic zone and incomplete nitrification, thus affecting the reaction performance (Bolek, n.d.). Ammonium nitrogen concentration also influences the ammonia-oxidizing bacteria community and hence biological nitrogen removal (Sui et al., 2014). Similarly, in bioreactor for energy generation applications, characteristics such as dry and nitrogenous content in the feedstock control the process efficiency as well as biogas quality (Lv et al., 2019). Those parameters often are determined through chemical-physical analysis. To ensure compliance with international standards as well as strict health and safety measures, specialized laboratories carry out those tests which can be very expensive (Bolek, n.d.; Government of Canada, 2017; *Proximate and Ultimate Analysis*, n.d.). Nonetheless, feedstock characterization is essential for process modeling and quality control. It serves as an input in reaction simulation software such as the activated sludge models (ASM), used in wastewater treatment plants (Henze et al., 2000).

1.2 Bioponics Case Study

Bioreactors have a wide range of applications and are a common alternative for waste revalorization. In bioreactors for bioponics applications, organic matter is used as an input to produce nitrate-rich liquid fertilizer that is suitable for hydroponics production (Khiari et al., 2019; Xie et al., 2022). Circulus AgTech is an example of a bioreactor based start up. Their product is an alternative to conventional synthetic fertilizer use which has huge environmental impacts. Indeed, with intensive agricultural practice, excess amounts of nutrients are often applied to crops leading to important air and water pollution. It also significantly contributes to global warming (US EPA, 2021). However, one of the issues that can arise is the production of potent greenhouse gasses such as nitrous oxide (N_2O) from the bioreaction. To ensure the economic viability of their solution, there also is a need to optimize nitrate production. To this end, feedstock composition plays a crucial role in influencing the system state and output. Moreover, in bioponics applications, feedstocks come from different sources and thus its characteristics vary from one batch to another (Khiari et al., 2019; Xie et al., 2022). Hence, conducting regular laboratory analysis for feedstock characterization can be economically straining and other methods for feedstock characterization should be considered.

1.3 Bioreactor process optimization

There have been several attempts to model the bioreactor process for optimization and control purposes. Research and development have notably been conducted to assess nitrification and denitrification processes through kinetics models. Shanahan & Semmens (2015) developed a model to assess the impact of pH and alkalinity on reaction kinetics in a membrane aerated bioreactor. It is based on Shanahan & Semmens' model, (2004). The simulation focused on nitrification and assumed denitrification was negligible. The controlled parameters include ammonium, nitrate, dissolved oxygen, hydronium ions, bicarbonate, dissolved carbon dioxide, dibasic hydrogen phosphate, monobasic hydrogen phosphate, and biofilm thickness; a single bacterial species was taken into consideration for ammonium to nitrate conversion. Validation and calibration were done using experimental data; the final simulation accurately predicted nitrification for a pH ranging from 5.5 to 8.5. Similarly, Al-Samawi & Shamkhi (n.d.) simulated nitrification in a stirred tank reactor at moderate temperature using SIMULINK®. Mathematical equations described both the microbial growth and the substrate removal to monitor dissolved oxygen and pH. Other process parameters involved are the temperature and the hydraulic retention time. A strong

correlation, R-squared = 0.946, was found between simulated and experimental data; slight variations could be attributed to the several assumptions made in the modeling process such as uniform ammonium distribution in the bioreactor, constant pH and temperature. Simultaneous nitrification and denitrification have also been explored to assess the impacts of dissolved oxygen, food/microorganism, C/N ratio and pH on the bioreaction. The Lawrence–McCarty model and a general mathematical model for a single-sludge wastewater treatment system were used for the simulation (He et al., 2009). Mathematical control theory models have also been developed to derive feedstock characteristics such as heterotrophic biomass, readily biodegradable soluble substrate or slowly biodegradable substrate from oxygen, water flows, total suspended solid and supplied air data (Hedegärd & Wik, 2011)

Although kinetics and control theory models can successfully model bioreactor processes and are currently the most commonly used extensively in process control. Though the development of those physical models requires deep knowledge of biochemical systems interactions (Mowbray et al., 2021) for a particular process they only need to be developed once. Moreover physical models are not practical to solve high-dimensional problems (Bensoussan et al., 2020) as their development is cost-intensive and time-consuming (Mowbray et al., 2021).

Consequently, data-driven approaches have seen an increased use in the past decades. Machine learning (ML) models are suitable for complex high-dimensional and non-linear biochemical models (Bensoussan et al., 2020; Mowbray et al., 2021). In bioreactor applications, machine learning has been used to establish relationships between operational conditions (pH, Dissolved oxygen, temperature, etc.), bacteria metabolism and product output. Certain algorithms can estimate system state in real-time state. Using an artificial neural network (ANN), (del Rio-Chanona et al., 2017) successfully evaluated process rate of change and future time steps in a feedback fermentation reactor for sugar and lysine production. The measured coefficient of determination R-squared was 0.997 (del Rio-Chanona et al., 2017; Mowbray et al., 2021). ML, however, cannot be used alone for model predictive control, unlike physical models. ML models can merely be used to predict data similar to what it has been trained on (Howard & Gugger, 2020), unlike physical models its predictions are not based on understanding of the system that is being predicted. This stems from the fact that ML is a data-based method. Furthermore, an ML model cannot be used to recommend an action but merely predict a result, in control, for example a model can predict/copy which of the controls procedures it has seen would likely be applied next it cannot itself find an optimal control if it has never been shown the optimal control applied to

the current case (Howard & Gugger, 2020). In response to this issue, hybrid models which combine physics-based model and machine learning have been explored for prediction and optimization (Bensoussan et al., 2020; Mowbray et al., 2021). In a fed-batch stirred tank reactor, system state at time *t* was an input to an artificial neural network, which then evaluated the reaction evolution at time *t*. Output from ANN was then transferred to a kinetics model to obtain the system state at time *t+1*. Such an approach reduced errors and optimized prediction in comparison with strict ANN models (Mowbray et al., 2021; Schubert et al., 1994). Similarly, here we propose the use of ML techniques in coordination with physical models to simplify input parameter characterization by developing ML techniques to recognize the relationships between the feedstock parameters and sensor readings while using existing physical models for model predictive control.

2. Design Approach

2.1 Design Criteria

Accuracy: Accuracy is a useful indicator to evaluate model performance; this metric is used to assess model effectiveness in recognizing correlations and patterns between variables in a dataset. Better predictions and insights would provide increased commercial value. The final design therefore needs to have an accurate model so that predictions may be trusted.

Speed: Speed is another crucial indicator for model performance. Fast predictions are essential to deliver added value to the customers, ML predictions are generally fast, however, training and development can take time, which needs to be considered in this project.

Scalability: Scalability is required for the solution to rapidly adapt to changes in a company's applications and system processing. The final model therefore needs to be easily adaptable to new requirements, or easily remade.

Ease of Use: The proposed model should be easily accessible by organizations. This is especially important for small companies like Circulus Agtech with less than a dozen employees and whose time for training employees may be limited. Therefore, any solution that is proposed needs to fit as frictionless as possible into the companies' existing structure. Data presentation should also be clear to be readily understood, evaluated and extracted. The final design therefore needs to be able to be easily integrated into companies current operations and ideally increase the usability of their current systems.

Cost: A main consideration is then to provide a low-cost solution. Again, bioreactor-based start-up companies such as Circulus Agtech may have limited economic resources. The final design therefore needs to be associated with the minimum capital and maintenance costs so that its benefits may outweigh its costs.

Reproducibility: Reproducibility refers to the potential for replication or duplication. As the current project is conducted as a feasibility study, the proposed design should be readily accessible by students, professionals and researchers to serve as a template for further research and development. The solution implemented should also serve as an inspiration/learning opportunity for users unfamiliar with machine learning, since these methods are useful but unfortunately often considered too complex to try.

2.2 Deep Learning

In this project, Deep Learning models are chosen to address the design considerations. Such models are more flexible than other, simpler, ML models which allows them to fit to arbitrarily complex functions (Howard & Gugger, 2020). Therefore, we consider them to be the best choice in terms of accuracy and potential model performance. The speed of running a deep learning model for prediction, similarly to any ML model is very fast. However, due to the large number of parameters included in deep learning models, data training is computationally expensive, hence we will limit the size of the deep models used in this project. This would enable not only a supercomputer but also a standard workstation to be capable of training and working with such a model. Ease of use is addressed in section 3.3 which describes how a deep model interface can be easily created to allow end-users to interact with the models. Such deep learning models are generally free, this is due to the availability of a multitude of free OpenSource programming libraries that may be used for their creation. Other associated costs that do not arise directly from deep learning are discussed in later sections of the report. Lastly, the use of deep learning methods is often regarded and considered as exceptionally difficult among ML methods. Therefore, the demonstration of a straightforward deep learning approach to solve relatively complex issues, serves as an inspiration to showcase the ease-of-use and underlying simplicity of using these methods; this would also allow this project to be used as a template for future application or reproduction.

The three principal deep learning approaches used here are the fully connected multi-layer-perceptron (MLP), the convolutional neural network (ConvNN) and the recurrent neural network (RNN). MLPs are the oldest versions of deep learning models and are based

on the perceptrons developed in the middle of the last century (Howard & Gugger, 2020), an MLP is considered deep if it has more than 1 hidden layer. MLPs are equivalent to ANNs, and have been applied to all kinds of ML problems, some examples are given in section 1.3. covering different modeling approaches. A ConvNN has an architecture that relies on the convolution operation, often likened to sliding a filter over a series or array of data, these models are typically used for machine vision applications as their filter or kernel allows them to extract relationships between different locations of the input data (Howard & Gugger, 2020). Here ConvNNs are used to extract temporal relationships from the input data through their kernels which we believe may improve their performance compared to MLPs. Lastly, RNNs were developed to utilize sequential data to make predictions, data such as text, and time-series as we are using here (Howard & Gugger, 2020), therefore they are the optimal choice for our project. We specifically use Gated Recurrent Units (GRUs) which are an improved version of RNNs that may be less computationally expensive than other RNN type models (Cho et al., 2014).

2.3. Selected Design

The design selected based on the above stated design criteria is to utilize deep learning models to predict the feedstock parameters from a time-series of measurements of process-monitoring sensors. This would reduce the number of parameters that need to be quantified analytically, reducing labor and equipment costs as well as time required to apply model predictive control for bioreactors with variable input feed.

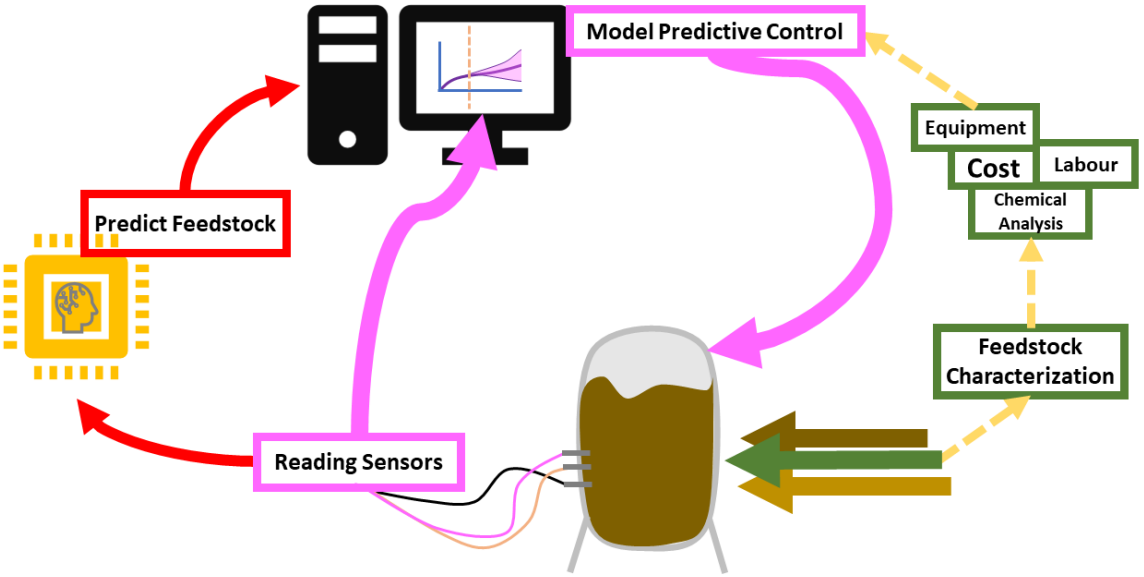

Figure 1: The proposed solution summarized, using Machine Learning for Feedstock Prediction to reduce the requirements of conventional Feedstock Characterization and streamline the use of Model Predictive Control for Bioreactors.

The inspiration for this design comes from Henze et al., (2000, p. 17) who note that some feedstock parameters may be derived from the behavior of the process immediately after the loading phase of the bioreactor. In current practice this requires knowledge of related feedstock parameters, the loading flow rate as well as the bioreactor volume. Therefore, since there does exist some relationship between the process behavior and the feedstock parameters, as complex as it may be, it can be approximated using deep neural networks (Howard & Gugger, 2020). Additionally, this example shows that in practice there may be a time-frame when no control is applied, i.e. during and immediately after loading, during which the sensor data would not be changed by control actions. This creates the possibility to use the sensor data during the initial "non-controlled" phase of the bioprocess to estimate the feedstock parameters.

The current project, therefore, was conducted to assess the feasibility of the proposed solution using a case-study. It will be assessed whether the mission statement "more accessible process control to facilitate commercial implementation of bioreactor-based technology" may be achieved by answering the following questions:

1. Can we predict the feedstock parameters using the recorded sensor data?
2. Which factors need to be considered when designing such a machine learning based system?

3. Which deep learning model architecture performs the best for the type of data used?
4. How difficult is the creation of a user interface to make this solution more accessible?
5. How may scarcity of labeled training data be overcome?

In general if an individual or organization wishes to the approach presented here then they generally can follow these steps:

1. Obtain an implementation of a physical model that represents your process (same as for standard model-based control)
2. Determine approximate range of feedstock parameters that could be expected in your application
3. Run the physical model multiple times with varying feedstock parameters as inputs and record the output variables that you have sensors for
4. Train a deep-learning-model using a time-series of this data that corresponds to a time-frame where no control needs to be applied.
5. Apply the deep-model with your sensor data from the same time-frame as it was trained for

The following section 3. Design Implementation will cover these implementation steps in detail, for a hypothetical case-study of a waste-water-fed, aerobic-batch-bioreactor application.

3. Design Implementation

3.1 Generating sample data

The Activated Sludge Model 1 (ASM1) (Henze et al., 2000) was selected to serve as the reference physical model due to its relative simplicity, and wide applicability, describing Nitrogen and Carbon dynamics, which are important to Waste-Water-Treatment and possibly also for Bioponics. The MATLAB® and Simulink® (MathWorks, n.d.) implementation of ASM1 that was developed to include pH and other ion calculations by Flores-Alsina et al., (2015) is used as the base implementation for our current project. The implementation was developed to monitor an entire Waste-Water-Treatment Plant with 2 anaerobic and 2 aerobic bioreactors. For this project we modified the ASM1 model to represent one aerobic batch bioreactor, as may be used during testing of a process (Xie et al., 2022). We focus on aerobic bioreactors specifically since they are particularly useful for producing liquid fertilizer (Xie et al., 2022). The simulation time for each run is set to 14 days, it approximates the time-range for which batch reactors are run by Xie et al., (2022), while also being the length of the short-run version of the original implementation (Flores-Alsina et al., 2015). The overall data

generation process is summarized in Figure 2. First the model was modified to be limited to one aerobic batch bioreactor. Then, from these runs, a selection of output variables would eventually be used as virtual sensor readings; this will be explained in more detail below.

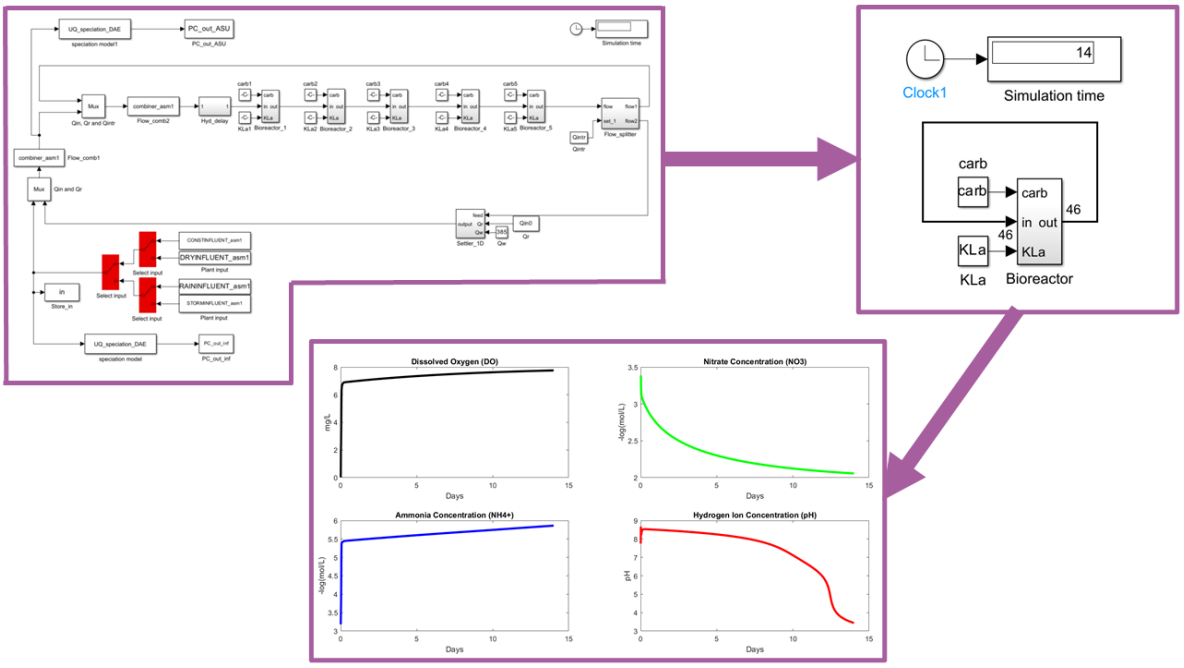

Figure 2: The data generation process is summarized. First the rather large ASM 1 Waste-Water-Treatment plant model is reduced to one aerobic batch bioreactor, then virtual sensor values are extracted from the model.

In order to robustly train any Machine Learning algorithm it is crucial to provide a descriptive amount of data. As deep learning architectures have thousands of parameters to be tuned they are not only capable of fitting to complex non-linear problems but also require more data so that all parameters are sufficiently tuned (Howard & Gugger, 2020). Here we first generated roughly 10,000 samples which were found through experimentation to give good performance as described later. In case that 10,000 samples would not have been enough 100,000 random possible input parameters were generated. Eight input parameters were selected from a list of feedstock parameters that cannot be assumed and are required to estimate other input parameters in the ASM1 model; this list was obtained from Henze et al., (2000, p. 22). The only parameters that were selected could be located both within the list and within the ASM1 model implementation by Flores-Alsina et al., (2015), as there exists no complete documentation of this implementation. Ranges for the input parameters were established using the minimum and maximum values of these parameters in the inflow data-sets of the ASM1 implementation. It was found that one of the originally chosen variables was always 0 for all times in all possible inflow files, this parameter was therefore

removed, the 7 remaining feedstock parameters as well as the ranges within which they were randomized are shown in Table 1 below. Input values, other than the 7 randomized parameters, were set to the values found for them in the constant inflow file which represents the mean values observed for the inflow during operation of the benchmark waste-water-treatment plant (Flores-Alsina et al., 2015).

Table 1: This table shows the feedstock parameters used to generate the sample data including their ranges. The "Input Range" describes the feedstock parameters for the input flow while the "Real Input Range" are the values that the SIMULINK® model received (i.e. Input Range + Previous Value in Bioreactor), we are predicting only the inflowing feedstock parameters so the values in the "Input Range" column. Only bold feedstock parameters were later used for prediction as explained in section 3.2. Model Development.

Feedstock Parameters:	Input Range:	Real Input Range:
Soluble Inert Org. Matter	[2, 42]	[30.1, 70.1]
Readily Biodegradable Substrate	[0, 160]	[3.1, 163.1]
Inert Suspended Org. Matter	[0, 400]	[1532.3, 1932.3]
Slowly Biodegradable Org. Matter	[9, 409]	[72.0, 472.0]
Soluble Ammonia Nitrogen	[1, 65]	[7.9, 71.9]
Soluble Biodegradable Org. Nitrogen	[0, 13]	[1.0, 14.0]
Slowly Biodegradable Org. Nitrogen	[0, 34]	[3.8, 37.8]

Units: mg/L

The model runs were conducted by simultaneously running 4 MATLAB® executables at the same time for 4-5 days each executable having a separate subset of the 100,000 possible input parameters to use as to only use each feedstock parameter combination once. Overall, 10922 files of 14 day simulations were created and used for model training and testing in the following sections. Each of the files consists of the virtual sensor values of dissolved oxygen (DO), nitrate ion concentration (NO_3^-), ammonium ion concentration (NH_4^+), and hydrogen ion concentration (H^+), each with a time-step of 15 minutes. Files are named according to the 7 feedstock parameter values used to generate that file in the same order as shown in Table 1. Virtual sensor data were not directly written into the files, rather 5% of Gaussian noise was added to each 'sensor' at each time-step (roughly 1000 samples received 10% of Gaussian noise to represent un-calibrated sensors and outliers), noise addition is done to ensure that the later developed deep-learning models will be useful in a real situation and not only in an idealized one.

The code from the data-generation is the only code that is not included in the project GitHub repository at the moment of writing (Harms & Crèche, 2022). It will be added eventually, however, the files will first need to undergo a cleaning of the code as well as the addition of some documentation on how to use them, which is not completed as of now.

3.2 Model Development

Firstly the generated files were loaded using Python (van Rossum, 1995) (in a Jupyter Notebook (Kluyver et al., 2016)) hosted on Google Colaboratory (*Google Colaboratory*, n.d.). One empty file was removed. A test and training set were created using random shuffling, to be able to evaluate the model performance on unseen data. The data was split into a training set of roughly 9000 training samples and 1900 testing samples (roughly a 82-18 split). The training and testing sets are important since we need to ensure that the model works not only on data it has seen but also "user-generated" or "real-world" data, in other words, using the training and testing sets we avoid overfitting (Howard & Gugger, 2020). Later we use 10% of the training data as a validation set (i.e. 900 samples for validation and 8100 as pure training data) during the actual model training process, this is used to ensure again that no overfitting occurs during training as well as it is used to tune the so-called hyperparameters or other parts of model structure that the model cannot learn by itself (Howard & Gugger, 2020). In essence, the validation set is what ultimately decides which model architecture will be evaluated against the test set by assisting the modeler in making the best design decisions (Howard & Gugger, 2020).

Each sample in the training, validation and testing sets consists of 4 columns containing sensor-variables (DO, NO_3^-, NH_4^+, H^+) each at 96 time-steps of 15 minutes (in total 24 hours). All samples were scaled using min-max scaling (Bhanja & Das, 2019):

$$x_{i,\,normalized} = \frac{x_i - max(x^{\{Train\}})}{max(x^{\{Train\}}) - min(x^{\{Train\}})} \tag{1}$$

For any value x_i of every variable x (i.e. DO, NO_3^-, NH_4^+ and H^+) for samples and each feedstock parameter for labels), here it is important to note that the maxima and minima of only the training data are used to scale the data in order to not carry over information from the testing set to the training set. For the purposes of scaling the combined training and

validation set is used, as we are generally aware of the maximum and minimum possible values especially for the feedstock parameters we could have also used the known maximum and minimum values for scaling. Scaling from 0-1 is done to make sure that all input variables as well as all output predictions are treated equally by the loss functions and not imply inherent importance of a variable by making its value larger.

After normalization a simple fully connected test model was created to establish whether the labels (the feedstock parameters) are possible to be predicted from the sensor values at all. The initial results (not shown) suggested that there were multiple parameters that could be predicted while there are others where the model was not able to find any correlation between the input and output data. Therefore, a simple statistical analysis was conducted to determine whether this problem is inherent to the data or whether it is the testing model that is at fault. Three Pearson correlation matrices were created using the means, range and standard deviation of the sensor inputs and the label values. Figure 3 shows the sensor means and label correlation matrix, in this as well as the other two matrices (not shown but may be accessed through the GitHub repository) showed that 5 out of the 7 label variables were somewhat correlated to the sensor values while 2 consistently showed correlations from 0.00-0.02. From this it was gathered that the sensors used would not likely be good predictors of the 2 uncorrelated feedstock parameters (Soluble Inert Organic Matter and Inert Suspended Organic Matter). It was later confirmed in experiments not shown that even more advanced model types were not able to find any correlation between these 2 feedstock parameters and the provided sensor inputs. Since both variables represent inert fractions of the feedstock it is reasonable that they do not have a significant effect on the process.

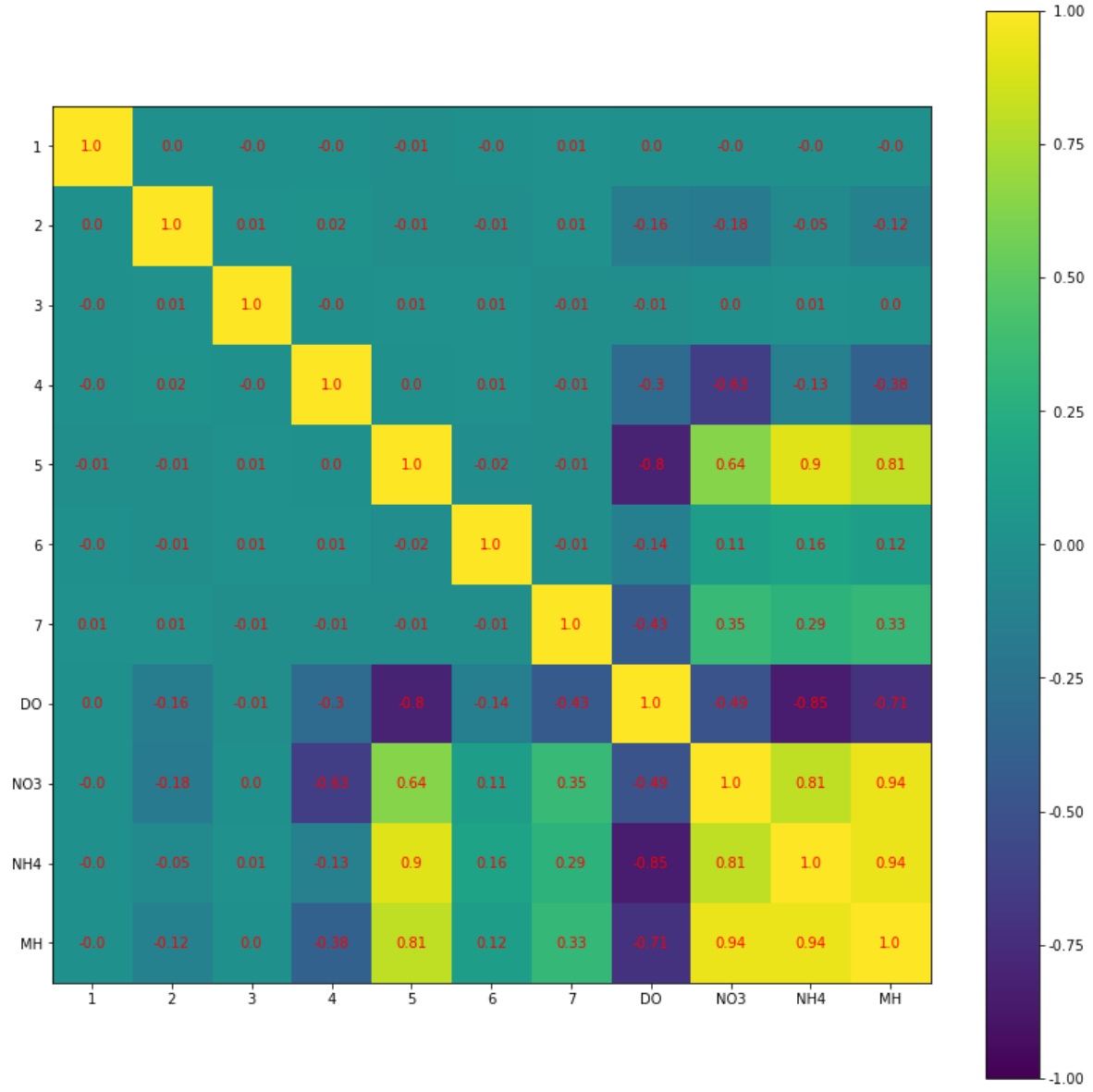

Figure 3: The correlation matrix between the feedstock parameters (1-7 as ordered in Table 1) and the sensor means (DO, NO_3^-, NH_4^+, H^+).

The deep learning models were developed using Tensorflow (Developers, 2022; Martín Abadi et al., 2015) and the Keras library (Chollet & others, 2015). In practice, however, the exact API and libraries used to develop the models do not make too much of a difference to the developer as most features are interchangeable between libraries as Howard & Gugger, (2020) point out. We chose Tensorflow and Keras for this project as it simultaneously allowed us to experiment easily with different model architectures, while being flexible enough to incorporate many specific design choices.

A fully connected multi-layer-perceptron (MLP), a convolutional neural network (ConvNN) and a recurrent neural network based on gated recurrent units (GRU) were designed. In table 2 the model architectures are summarized. Each model has roughly 50000 learnable parameters. All models were stochastically trained using an adaptive moment estimation optimizer (ADAM) (Kingma & Ba, 2017), with a maximum learning rate of 0.01, and mini-batches of 100 samples. The MLP and ConvNN models were trained for 200 epochs whereas the GRU only for 100 epochs as it was observed that the loss on the validation and training sets began to diverge if we increased this value. The loss function for the MLP and ConvNN was Mean Square Error (MSE) while for GRU the Mean Absolute Error (MAE) was used. The setting of these parameters as well as the specifics of the model architectures was based on trial-and-error experimentation. After training when validation loss fell into an acceptable range the performance was evaluated on the test set. It is important to note that since the training process is stochastic that not every run with the same parameters leads to the exact same training trajectory, so the models were, on occasion, re-compiled and re-trained to ensure that the optimization using the current parameters was indeed the best possible.

Table 2: This table summarizes the model architectures as well as the final validation loss for each model (Notice: the loss units are not all the same).

MLP model:	**ConvNN model:**	**GRU model:**
Dense, Units: 128	Conv2D, Units: 32	GRU, Units: 128
Dense, Units: 32	Flatten, Output size: 1504	Dense, Units: 32
Dense, Units: 32	Dense, Units: 32	Dense, Units: 5
Dense, Units: 5	Dense, Units: 32	
	Dense, Units: 5	
Training: 200 Epochs	Training: 200 Epochs	Training: 100 Epochs
Validation loss: 0.0092(MSE)	Validation loss: 0.0099 (MSE)	Validation loss: 0.0551(MAE)

3.3 Graphical User Interface

Having a Graphical User Interface (GUI) for applications meant to serve non-programmers is crucial for our project to showcase our software. Furthermore, a GUI also makes code much more accessible, especially when hosted as a web-application. Websites, however, consist of JavaScript and HTML, which may make their creation difficult for programmers that only have a background in Python or other programming languages commonly used for Data-Science. However, libraries and API's exist that allow Python

programmers to create web-based applications using only Python. Similarly, hosting also does not have to be expensive, plenty of Open-Source website hosting can be done using platforms like GitHub Pages (*GitHub Pages*, n.d.), or Hugging Face Spaces (*Spaces - Hugging Face*, n.d.). For our work we use the Streamlit (*Streamlit*, n.d.) for development, which is a simple-to-use API to create beautiful interfaces. Furthermore, Streamlit also has its own hosting service which is free for Open-Source projects that are stored on GitHub, which is ideal for our project.

In another case, one may also create a UI using Gradio (Abid et al., 2019) and host it on Hugging Face Spaces which can then be used as an API in addition to a UI. Therefore this way the interface may not only be used by people manually but also integrated into professional websites and provide model predictions automatically to any application, making it easy to integrate these models into the workflow of businesses, professionals and institutions. It should be noted that either option may incur costs and no longer remain a free service if used for commercial purposes. We choose the Streamlit option as it provides the most user-friendly interface, and keeps our work accessible to everyone. Ideally we would have liked to do both, since development of a simple Gradio interface would not be too difficult, and be more useful in a commercial setting. However, due to time constraints the Gradio interface is not yet completed and deployed at the time of writing and in this text only the Streamlit application will be discussed. After the writing of this report the Gradio interface will still be implemented, updates will be posted on the GitHub repository of this project.

4. Results and Discussion

4.1. Model Performance

The developed model performances for predicting the selected feedstock parameters are shown below in Table 3 and Table 4. Table 3 summarizes the performance over all variables, Table 4 gives a more detailed view of which models performed best on every feedstock parameter individually. Generally, the prediction of the 4th parameter, Soluble Biodegradable Organic Nitrogen (SBON), was the only parameter not predicted well by the models. Overall model performance at a mean R-squared value of about 0.89-0.9 for all models suggests that the prediction of feedstock parameters is largely successful. Moreso, considering the fact that all models are able to achieve an R-square above 0.97 for 4 out of 5 variables to be predicted it becomes apparent that these models are working exceptionally

well. The unfortunately low fit for the SBON parameter may be connected with the comparatively small input range for which samples were generated (see: Table 1). All SBON data is limited to a range of 13 mg/L whereas the ranges for the other variables are 2.6 to 30.8 times as large (34 mg/L to 400 mg/L). Taking this into consideration it suggests that the selection of important process variables and appropriate ranges that cover not only the required input range but also allow the model to extract the underlying relationship is a crucial step in applying the proposed design. If the input parameters and ranges are chosen properly then we see from these results that Deep Learning models such as the ones presented here may indeed be very useful and provide a simple method of estimating feedstock parameters that is highly accurate.

Table 3: The combined performance indicators for each model on the testing data. The best values are bold.

Model:	1	2	3	4	5
MLP	0.986	0.994	**0.980**	**0.564**	0.972
ConvNN	0.985	0.994	**0.980**	0.545	0.973
GRU	**0.989**	**0.996**	**0.980**	0.524	**0.980**

Table 4: The R-squared of each model for each of the predicted variables. 1. Readily Biodegradable Substrate; 2. Slowly Biodegradable Organic Matter; 3. Soluble Ammonium Nitrogen; 4. Soluble Biodegradable Organic Nitrogen; 5. Slowly Biodegradable Organic Nitrogen. The best values are in bold.

Model:	Explained Variance:	Mean R^2:
MLP	**0.8990**	**0.8992**
ConvNN	0.8950	0.8954
GRU	0.8928	0.8938

4.2 Graphical User Interface

Our Streamlit GUI serves as a platform for users to experiment with the trained deep-learning models described in this text as well as an example of how a GUI may be easily developed in general.

Several features were implemented to ensure user's satisfaction and ease of use. After uploading its file with the four sensor readings data (Concentration of hydrogen ion, dissolved oxygen, nitrate and ammonium). To ensure the UI is compatible with various file arrangements, the user can identify the position of each sensor data in his file with a number input widget (Figure 4). They can also specify if its document contains a header through the checkbox feature. After proper settings have been established, he will have the possibility to display its dataset in a table, to visualize concentration of all four parameters as function of the time (Figure 5). Finally by scrolling down he can instantly access feedstock characteristics predictions for the MLP, ConvNN and GRU models.

For learning purposes, the UI also is accessible without having a file containing sensor readings data. In such a case, the user could test one of the sixth examples provided using a scroll down menu. They can visualize the tabulated and graphed data. Finally, for the example documents, the user could compare experimental results with the three models' predictions as they would both be displayed (Figure 6).

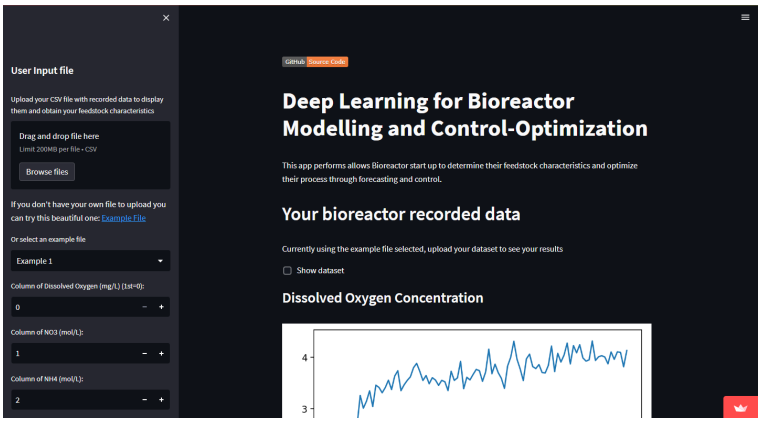

Figure 4. Features of the Deep Learning for Bioreactor Modelling and Control-Optimization UI

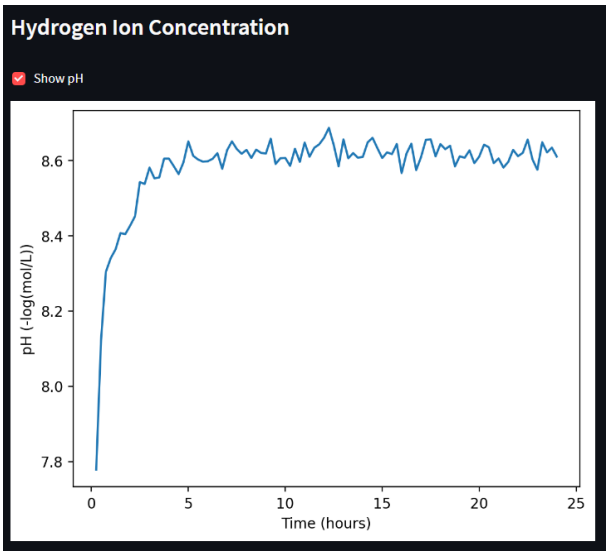

Figure 5. pH time-series plot from the Deep Learning for Bioreactor Modelling and Control-Optimization UI

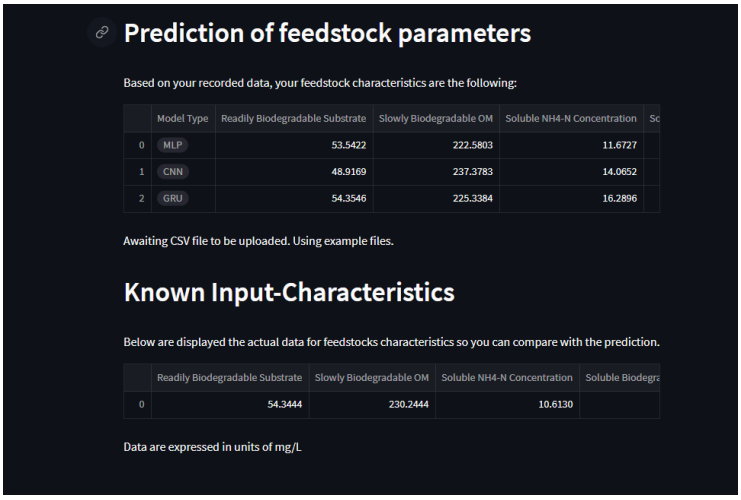

Figure 6. Prediction of feedstock parameters and known input characteristics from the Deep Learning for Bioreactor Modelling and Control-Optimization UI

4.3 Design Considerations

4.3.1 Environmental Considerations

The feasibility project demonstrated the potential of machine learning for feedstock characterization. Coupled with a process control system, it is expected that such a solution could significantly improve bioreactor process while reducing undesired outputs. For a company like Circulus AgTech, understanding the relationship between feedstock composition and data sensor readings could optimize nitrate production while reducing nitrous oxide emissions. Indeed, experimental studies evidenced effects of operating

conditions such as dissolved oxygen on nitrous oxide emissions during nitrification and denitrification (Aboobakar et al., 2013; Tallec et al., 2006; Wen et al., 2020) Based on such information Circulus AgTech could first determine which feedstock is the most appropriate for an optimized output. This would allow process stability and constant output quality which are essential for scalability and commercialization. Then, combining machine machine learning and physics based optimization and control set up could improve bioreactor performance and limit nitrous oxide emissions.

Although machine learning is promising for environmental protection in bioreactor applications, large data centers, which may be used to train machine learning models, require an important amount of energy and water resources for cooling systems and thus they emit greenhouse gasses (*The Impact of Data Centers on The Environment*, n.d.). In the current project the training, though time intensive, model training does not strictly require data-centers for training but is achievable on a personal computer, though the availability of a Graphics Processing Unit (GPU) would be useful.

4.3.2 Economic consideration

The designed machine learning models did not require any capital cost investment. However, it should be noted that if used for commercial purposes, development of similar tools would incur costs and no longer remain a free service. Labor costs would be associated with the architecture elaboration. Then, to ensure accuracy in the model prediction, sufficient data should be collected. To this end, a kinetics based model, such as ASM1 used in this feasibility study, would be required to simulate bioreaction and generate the data. If the physics-based simulation is conceived on the Matlab programming platform, additional costs would be induced to purchase a Mathwork license for corporate organizations. For a single user, prices start at USD$1200 and would increase for multiple-user access (*Pricing and Licensing*, n.d.).

As mentioned previously, a combination of machine learning model and control features to evaluate feedstock composition and optimize bioreaction would support companies' economic growth.

4.3.3 Social Considerations

The proposed design offers several social benefits. It can serve as a learning tool for students, professionals and researchers. Indeed, Model and GUI files and codes are available on GitHub in order to enable readers of this report to use this project as a template for their

own work. The specific version of the code referred to in this report is archived and citable (Harms & Crèche, 2022). The MIT license was attached to all code so that it may be used even for commercial purposes, the specific conditions of the license are specified here (*MIT License*, 2022). Moreover, to favor access to such intellectual resources, the application has been deployed on two platforms: Streamlit cloud, and the data-science hosting service Huggin Face.

Careful thoughts were given to the UI design to help users in their task completion. Each interface component is simple to use, access, and comprehend; information architecture, interaction design, and graphic design are consistent and predictable to meet user's expectations and ensure their satisfaction. As mentioned previously, the interface can be easily incorporated into the workflow of enterprises, professionals, and institutions. It is then an asset for bioreactor-based companies such as Circulus AgTech.

When designing a machine learning based model, strong attention should also be given to data ethics which "encompasses the moral obligations of gathering, protecting, and using personally identifiable information and how it affects individuals." (*5 Principles of Data Ethics for Business*, 2021). While gathering personal data, ethical principles must be followed: ensuring individuals agree to share their personal information, being transparent about the end use of the data, maintaining data privacy, collecting only required data and understanding limits and bias (*5 Principles of Data Ethics for Business*, 2021). In this feasibility project, we self-generated the data, hence we did not encounter privacy issues. However, it should be noted that as all models have limits, thus the final design should not be used beyond the data range it was trained for.

5. Future Improvements

In future work it would be important to test not only the theoretical feasibility of such a feedstock prediction system but also quantify how well such models perform in a practical setting. Unfortunately this was not possible due to the timing of this project. Other improvements include the addition of a Gradio API/UI as mentioned above that will showcase how interfaces not only for manual use but also for automatic prediction may be used, this will be added to this fork of the GitHub repository hosting the Streamlit GUI of this project. Additionally, in future work, a more thorough hyperparameter tuning would be necessary to be certain that the developed models are performing at the best possible level, rather than the more informal experimental approach used here, again due to time constraints.

6. Conclusion

Feedstock characterization is essential in bioreactor optimization and control; however their characterization is costly and thus not practical for small-scale companies. The feasibility study assesses the potential of machine learning for feedstock characterization and bioreaction optimization. Tests were conducted through three deep learning architectures: a fully connected Multi-Layer-Perceptron model, a Convolutional Neural Network and a Recurrent Neural Network using Gated-Recurrent Units (GRUs). Each model successfully predicted feedstock characteristics, the mean R-square over all models and all variables predicted was 0.8961. The GRU model achieved the highest coefficient of determination on 4 out of 5 predicted feedstock parameters. Hence machine learning proved its potential for improved bioreactor process reaction. A graphical user interface was also developed to host the models and allows professionals, students or researchers to use it as a learning opportunity.

7. References

5 Principles of Data Ethics for Business. (2021, March 16). Business Insights Blog. https://online.hbs.edu/blog/post/data-ethics

Abid, A., Abdalla, A., Abid, A., Khan, D., Alfozan, A., & Zou, J. (2019). *Gradio: Hassle-free sharing and testing of ML models in the wild*. https://doi.org/10.48550/arXiv.1906.02569

Aboobakar, A., Cartmell, E., Stephenson, T., Jones, M., Vale, P., & Dotro, G. (2013). Nitrous oxide emissions and dissolved oxygen profiling in a full-scale nitrifying activated sludge treatment plant. *Water Research*, *47*(2), 524–534. https://doi.org/10.1016/j.watres.2012.10.004

Al-Samawi, A. A., & Shamkhi, M. S. (n.d.). Model Development and Simulation of Nitrification in SHARON Reactor in Moderate Temperature by Simulink. *Volume 3 Issue 10–October 2014*, *99*(109), 48.

Bensoussan, A., Li, Y., Nguyen, D., Tran, M.-B., Yam, S., & Zhou, X. (2020). *Machine Learning and Control Theory*.

Bhanja, S., & Das, A. (2019). *Impact of Data Normalization on Deep Neural Network for Time Series Forecasting* (arXiv:1812.05519). arXiv. https://doi.org/10.48550/arXiv.1812.05519

Bolek, R. (n.d.). *Determination of Readily Biodegradable COD (rbCOD)*. 43.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. https://doi.org/10.48550/arXiv.1406.1078

Chollet, F. & others. (2015). *Keras*. GitHub. https://github.com/fchollet/keras

del Rio-Chanona, E. A., Fiorelli, F., Zhang, D., Ahmed, N. R., Jing, K., & Shah, N. (2017). An efficient model construction strategy to simulate microalgal lutein photo-production dynamic process. *Biotechnology and Bioengineering*, *114*(11), 2518–2527. https://doi.org/10.1002/bit.26373

Developers, T. (2022). *TensorFlow*. Zenodo. https://doi.org/10.5281/zenodo.6574269

Flores-Alsina, X., Kazadi Mbamba, C., Solon, K., Vrecko, D., Tait, S., Batstone, D. J., Jeppsson, U., & Gernaey, K. V. (2015). A plant-wide aqueous phase chemistry module describing pH variations and ion speciation/pairing in wastewater treatment process models. *Water Research*, *85*, 255–265. https://doi.org/10.1016/j.watres.2015.07.014

GitHub Pages. (n.d.). GitHub Pages. Retrieved December 3, 2022, from https://pages.github.com/

Google Colaboratory. (n.d.). Retrieved December 3, 2022, from https://colab.research.google.com/

Government of Canada, P. S. and P. C. (2017, March 29). *Average costs for the laboratory analysis of a sample—Guidance and Orientation for the Selection of Technologies—Contaminated sites—Pollution and waste management—Environment and natural resources—Canada.ca*. https://gost.tpsgc-pwgsc.gc.ca/fld_cst.aspx?lang=eng#2

Harms, J. Z., & Crèche, A. (2022). *joelz575/DeepBioreactorModelling: V1.0.0-alpha*

(v1.0.0-alpha). Zenodo. https://doi.org/10.5281/ZENODO.7395353

He, S., Xue, G., & Wang, B. (2009). Factors affecting simultaneous nitrification and de-nitrification (SND) and its kinetics model in membrane bioreactor. *Journal of Hazardous Materials*, *168*(2–3), 704–710.

Hedegärd, M., & Wik, T. (2011). An online method for estimation of degradable substrate and biomass in an aerated activated sludge process. *Water Research*, *45*(19), 6308–6320. https://doi.org/10.1016/j.watres.2011.09.003

Henze, M., Gujer, W., Mino, T., & van Loosdrecht, M. C. M. (2000). *Activated sludge models ASM1, ASM2, ASM2d and ASM3* (reprint). IWA Publishing.

Howard, J., & Gugger, S. (2020). Fastai: A Layered API for Deep Learning. *Information*, *11*(2), Article 2. https://doi.org/10.3390/info11020108

Khiari, Z., Kaluthota, S., & Savidov, N. (2019). Aerobic bioconversion of aquaculture solid waste into liquid fertilizer: Effects of bioprocess parameters on kinetics of nitrogen mineralization. *Aquaculture*, *500*, 492–499.

Kingma, D. P., & Ba, J. (2017). *Adam: A Method for Stochastic Optimization* (arXiv:1412.6980). arXiv. http://arxiv.org/abs/1412.6980

Kluyver, T., Ragan-Kelley, B., P&#233, Rez, F., Granger, B., Bussonnier, M., Frederic, J., Kelley, K., Hamrick, J., Grout, J., Corlay, S., Ivanov, P., Avila, D., n, Abdalla, S., Willing, C., & Team, J. D. (2016). Jupyter Notebooks – a publishing format for reproducible computational workflows. *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, 87–90. https://doi.org/10.3233/978-1-61499-649-1-87

Lv, Z., Chen, Z., Chen, X., Liang, J., Jiang, J., & Loake, G. J. (2019). Effects of various feedstocks on isotope fractionation of biogas and microbial community structure during anaerobic digestion. *Waste Management*, *84*, 211–219. https://doi.org/10.1016/j.wasman.2018.11.043

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian

Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Jia, Y., Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, … Xiaoqiang Zheng. (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. https://www.tensorflow.org/

MathWorks. (n.d.). *MathWorks—Makers of MATLAB and Simulink*. Retrieved December 4, 2022, from https://www.mathworks.com/

MIT License. (2022, September 26). Choose a License. https://choosealicense.com/licenses/mit/

Mowbray, M., Savage, T., Wu, C., Song, Z., Anye Cho, B., del Rio-Chanona, E., & Zhang, D. (2021). Machine learning for biochemical engineering: A review. *Biochemical Engineering Journal*, *172*, 108054. https://doi.org/10.1016/j.bej.2021.108054

Pricing and Licensing. (n.d.). Retrieved December 4, 2022, from https://www.mathworks.com/pricing-licensing.html

Proximate and Ultimate Analysis. (n.d.). SGSCorp. Retrieved December 3, 2022, from https://www.sgs.com/en-us/services/proximate-and-ultimate-analysis

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. https://doi.org/10.1016/0168-1656(94)90189-9

Shanahan, J. W., & Semmens, M. J. (2004). *Multipopulation model of membrane-aerated biofilms—PubMed*. https://pubmed.ncbi.nlm.nih.gov/15224752/

Shanahan, J. W., & Semmens, M. J. (2015). Alkalinity and pH effects on nitrification in a membrane aerated bioreactor: An experimental and model analysis. *Water Research*, *74*, 10–22.

Spaces—Hugging Face. (n.d.). Retrieved December 3, 2022, from https://huggingface.co/spaces

Streamlit • The fastest way to build and share data apps. (n.d.). Retrieved December 3, 2022, from https://streamlit.io/undefined/

Sui, Q., Liu, C., Dong, H., & Zhu, Z. (2014). Effect of ammonium nitrogen concentration on

the ammonia-oxidizing bacteria community in a membrane bioreactor for the treatment of anaerobically digested swine wastewater. *Journal of Bioscience and Bioengineering*, *118*(3), 277–283. https://doi.org/10.1016/j.jbiosc.2014.02.017

Tallec, G., Garnier, J., Billen, G., & Gousailles, M. (2006). Nitrous oxide emissions from secondary activated sludge in nitrifying conditions of urban wastewater treatment plants: Effect of oxygenation level. *Water Research*, *40*(15), 2972–2980. https://doi.org/10.1016/j.watres.2006.05.037

The Impact of Data Centers on The Environment. (n.d.). Retrieved December 4, 2022, from https://crewdle.com/blog/the-impact-of-data-centres-on-the-environment

US EPA, O. (2021, March 22). *Understanding the Impacts of Synthetic Nitrogen on Air and Water Quality Using Integrated Models* [Overviews and Factsheets]. https://www.epa.gov/sciencematters/understanding-impacts-synthetic-nitrogen-air-and-water-quality-using-integrated

van Rossum, G. (1995). *Python reference manual* (R 9525). Article R 9525. https://ir.cwi.nl/pub/5008

Wen, J., LeChevallier, M. W., Tao, W., & Liu, Y. (2020). Nitrous oxide emission and microbial community of full-scale anoxic/aerobic membrane bioreactors at low dissolved oxygen setpoints. *Journal of Water Process Engineering*, *38*, 101654. https://doi.org/10.1016/j.jwpe.2020.101654

Xie, Y., Spiller, M., & Vlaeminck, S. E. (2022). A bioreactor and nutrient balancing approach for the conversion of solid organic fertilizers to liquid nitrate-rich fertilizers: Mineralization and nitrification performance complemented with economic aspects. *Science of The Total Environment*, *806*, 150415. https://doi.org/10.1016/j.scitotenv.2021.150415