Carnegie Mellon University

CARNEGIE INSTITUTE OF TECHNOLOGY

REPORT

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF Master of Science

TITLE Using Deep Learning to make real-time predictions of Glycosylation in Monoclonal Antibodies

PRESENTED BY Deepro Banerjee

ACCEPTED BY THE DEPARTMENT OF

Chemical Engineering

ANNE S. ROBINSON, DEPARTMENT HEAD AND ADVISOR 12/9/2019 DATE

Lynn Walker

LYNN WALKER, PROFESSOR AND GRADUATE PROGRAMS CHAIR 12/9/2019 DATE

Contents

Table of Figures

Motivation

The Global Pharmaceuticals Market

The global pharmaceuticals market reached $1.2 trillion in 2018, up $100 billion from 2017, according to the Global Use of Medicines report from the IQVIA Institute for Human Data Science. For the US specifically, the 2018 spending was $485 billion, up 5.2% over the previous year. With significant transformation in the laboratory and in strategy, technology and operations, biopharmaceuticals could become the core of the pharmaceutical industry [1]. Recently, there has been a rapid advancement in the number of biopharmaceuticals available for clinical use. The United States Food and Drug Administration (USFDA) has approved 18 new biopharmaceutical products in 2012, which include eight recombinant proteins, two monoclonal antibodies and one engineered antibody-like biomolecule, covering a wide range of innovation, novelty, healthcare and market impact. Research data released by PharmaVision in 2009 shows that biopharmaceuticals represent 7.5% of all drugs in the market and account for almost 10% of the total expenses for marketed drugs. The usage of biopharmaceuticals are increasing at the rate greater than 20% per year. They are already being used 74% more than chemically derived pharmaceuticals in life-saving situations. Moreover, biopharmaceuticals cover greater than 30%

of all pipeline research programs going-on at present. Figure 1 below shows the Global Biopharmaceutical Market Size and Forecast over the period of 10 years from 2015 to 2024.

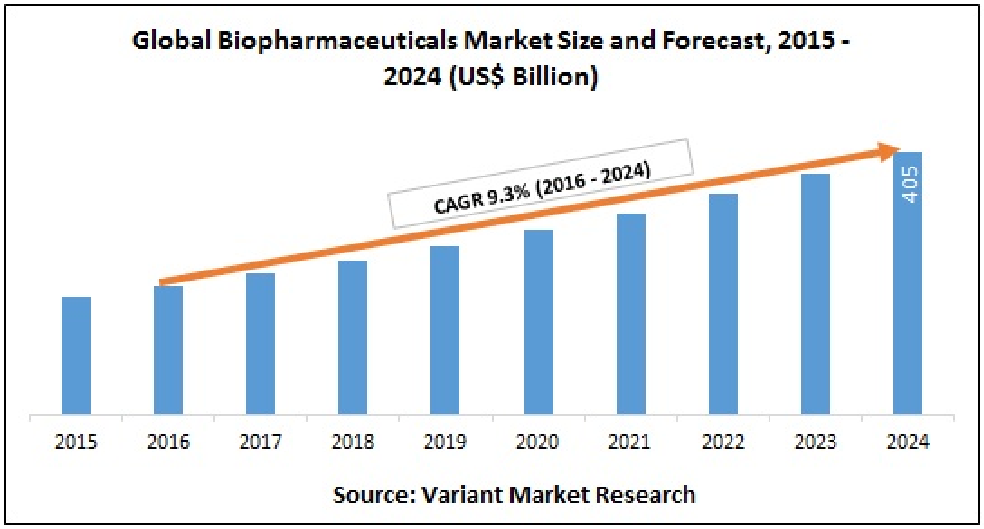
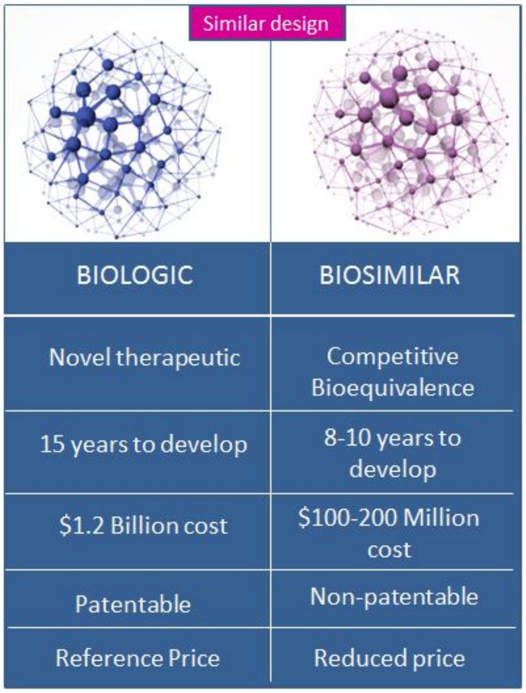


Figure 1: Global Biopharmaceutical Market Size and Forecast

Monoclonal Antibodies and the advent of Biosimilars

Monoclonal Antibodies (mAbs) represent the largest fraction of marketed biopharmaceutical products and those in clinical development [2]. In 2018, the total global sales of mAbs exceeded US $122 billion, indicating the high commercial value of these therapeutics. In spite of having several therapeutic benefits, these biologics pose an economic challenge for the healthcare system as elevated drug prices are a problem for patients and policy makers alike. The development of biosimilars has grabbed increasing attention with the expiration of exclusive patents of many high profile and blockbuster monoclonal antibody products. The FDA defines a biosimilar as “a biological product that is approved based on a showing that it is highly similar to an FDA-approved biological product, known as a reference product, and has no clinically meaningful differences in terms of safety and effectiveness from the reference product.”. Introducing biosimilars will greatly affect global sales of monoclonal antibody products as biosimilar monoclonal antibodies are approved in areas that cannot currently acquire costly innovator products [3].



Similar design

BIOLOGIC	BIOSIMILAR
Novel therapeutic	Competitive Bioequivalence
15 years to develop	8-10 years to develop
$1.2 Billion cost	$100-200 Million cost
Patentable	Non-patentable
Reference Price	Reduced price

Figure 2: Comparison between Biologics and Biosimilars

Characterization of Critical Quality Attributes

Regulatory agencies are pursuing extensive characterization of the complete quality attributes of the drug, since biosimilars duplicate the amino acid sequence in the innovator molecule, differing only in quality profiles. Although there is an emergence of biosimilars, focus has shifted from improving productivity, greatly increased alongside host cell optimization, to providing uniform product quality across batches, thereby improving profitability altogether. The 'Quality by design' framework has been implemented in the past decade, by the biopharmaceutical industry, which has quality related to drug product safety and efficacy built into every stage of the process (following ICH Q8 guidelines). The primary goal of the biopharmaceutical industry is therefore, maintaining drug product quality and ensuring product safety and efficacy [4].

N-Glycosylation in Monoclonal Antibodies

One of the most important determinants of mAb quality is N-glycosylation, a posttranslational modification of the antibody in which an oligosaccharyltransferase complex in the endoplasmic reticulum adds a sugar substrate (glycan) to the Asn-X-Ser/Thr motif in the heavy chain of the mAb (where X is any amino acid other than Pro). As the mAb traverses the Golgi complex, the attached oligosaccharide is subjected to a series of non-template driven enzymatic modifications mediated by the localized glycosyltransferase enzymes in the different Golgi compartments. The intricate dynamics of multiple glycosyltransferase enzymes determine the eventual fate of the core glycan and result in the formation of a diverse array of glycan isoforms that affect the immunogenicity, effector functions, and the pharmacokinetic properties of the mAb, and consequently the final drug product quality [5]. As a result, manufacturers are driven towards understanding, characterizing and regulating the glycoform distribution in mAbs, so that a uniform glycan profile is maintained and quality standards provided by regulatory agencies worldwide are met with. Therefore, the overall objective of this thesis is to provide a rational framework to model and estimate glycosylation in monoclonal antibodies produced in mammalian cells.

Monoclonal Antibodies and N-Glycosylation

Antibodies

Antibodies or Immunoglobins are proteins produced by the immune system in response to a foreign substance called an antigen. When alien substances like bacteria, viruses or other harmful invaders enter the human body, the immune system is able to recognize it as foreign because molecules on the surface of the antigen differ from those found in the body. To eliminate the invader, the immune system carries out a number of mechanisms, including one of the most important, antibody production. Post production, antibodies circulate, attack and neutralize antigens that are identical to the one that triggered the immune response by binding to them. Different antibodies adopt different mechanisms to neutralize the threat. Some antibodies like antitoxins can neutralize the poison simply by changing its chemical composition. Other antibodies can immobilize the invading microbes by attaching themselves to it. In other cases, the antigen is subjected to a chemical chain reaction by a complement system, which is a series of proteins found in the blood. Either the reaction triggers the lysis (bursting) of the invading microbe or it attracts microbe-killing scavenger cells that ingest, or phagocytose, the invader. Once it starts, antibody production continues for several days until all antigen molecules are removed. Antibodies remain in circulation for several months, providing extended immunity against that particular antigen. The core structure of these antibodies consists of two identical heavy chains and two identical light chains that bind together to form a Y-shaped structure as shown in Figure 3 below:

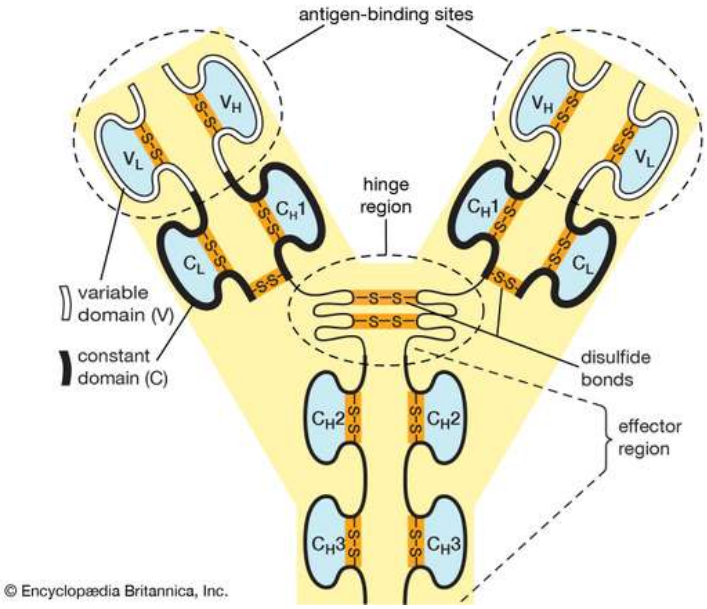


Figure 3: Antibody Structure

For a particular class of antibody, the stem and the bottom part of the arm is similar and it is called the constant region. The top of the Y shape contains the variable region that bind to antigens. Each antibody has two identical antigen-binding sites, one at the end of each arm, and the antigen-binding sites vary greatly among antibodies. Depending upon their constant region, antibodies are grouped into five classes. They are IgG, IgM, IgA, IgD, and IgE. The classes differ not only in their constant region but also in their activity. For example, IgG, the most common antibody, is present mostly in the blood and tissue fluids, while IgA is found in the mucous membranes lining the respiratory and gastrointestinal tracts [6].

Monoclonal Antibodies

Monoclonal antibodies (mAbs) are antibodies that are produced by identical immune cells originating from a single parent cell and have a very high specificity for a particular antigen. There are different forms of mAbs, but variations of the IgG1 form, which consists of two heavy chains and two light chains, are most widely used as therapeutic proteins. The four polypeptide chains of an IgG1 form two antigen binding or variable regions and a single constant region.

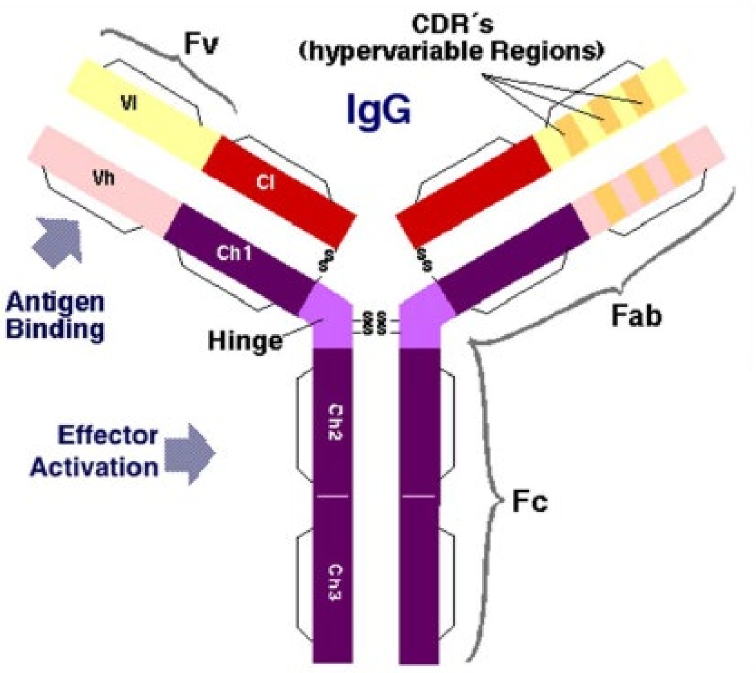


Figure 4: Monoclonal Antibody Structure

The variable region allows the monoclonal antibody high specificity to the antigen whereas the constant region is involved in determining the mechanism that will be used to destroy the antigen. The specificity for an antigen makes monoclonal antibodies ideally suited to tackle a variety of diseases ranging from rheumatoid arthritis, multiple sclerosis, heart disease and cancer [7].

Monoclonal Antibodies expression systems

Therapeutic mAbs are mainly synthesized in mammalian host cell lines such as Chinese Hamster Ovary (CHO) cells, NS0 murine myeloma cells, PER.C6® human cells. Over half of all currently approved mAbs are produced in Chinese Hamster Ovary (CHO) cell lines (data accessed online from “Drugs@FDA”). The preference for CHO as the host expression system for recombinant therapeutic proteins arises due to several reasons. Regulatory agencies have greater confidence in the safety of CHO-based therapeutic products due to years of research and safety testing that has been carried out on this commercial cell line. From a manufacturing perspective, the availability of powerful gene amplification systems results in higher yield of recombinant proteins thus improving overall profitability. Currently, antibody titers from CHO cell culture have reached up to 1 g/L for batch cultures and 1-10 g/L for fed-batch cultures, which is a 100-fold improvement over similar process in the 1980s, indicating the extensive development that has taken place in this field. Additionally, CHO cells are capable of adapting to suspension cultures that are ideal for large-scale glycan production making them a preferred host for most therapeutics. In addition to that, CHO cells allow post-translational modifications to recombinant proteins that enables the formation of structures commonly observed in human cells, thereby ensuring biocompatibility between the products manufactured in CHO cells and human beings [5].

Critical Quality Attributes of Monoclonal Antibodies

Although MAbs can effectively treat several diseases, high manufacturing costs make these therapeutics prohibitively expensive making them unaffordable for the majority of population around the globe. Biosimilars could reduce the costs associated with monoclonal antibody therapeutics and make these treatments more accessible. The advent of these biological drugs, along with the rapid advancement in analytical characterization techniques, has increased the focus on evaluating and recognizing the factors that affect the quality attributes of antibodies. In the quality by design paradigm, only those physical and chemical changes that affect the safety and efficacy of the drug are labelled as critical quality attributes (CQAs), whose levels should be maintained within predefined ranges. A few quality attributes of monoclonal antibodies, known to influence antibody activity are discussed below [4]:

- Aggregation – Aggregated proteins must be removed from the final drug product by applying appropriate downstream purification strategies since they are known to induce adverse immunological responses in patients and reduce product yield. It occurs due to exposed hydrophobic patches on the protein or changes in operating conditions.
- Glycation – Glycation is the bonding of a reducing sugar to the amine group of lysine side chains without enzymatic regulation. Glycation of antibodies can take place during cell

culture or in vivo upon storage with lactose. The resulting glycated antibody shows lowered immunoreactivity and increases the drug product heterogeneity.

- Cysteine variants – Monoclonal antibodies contain interchain and intrachain disulphide bonds, which, if disturbed, can cause heterogeneity and disulphide bond scrambling in mAbs. Reduced mAb potency has been observed due to the presence of incomplete disulphide pairing on the protein.

Apart from these attributes, glycosylation, is considered one of the most critical determinant of protein quality and is discussed here in brief.

N-linked Glycosylation

This thesis focusses on N-linked glycosylation, which entails the transfer of a core pentasaccharide from a donor molecule to the recipient protein within the endoplasmic reticulum (ER) of the host cell. On attachment, monosaccharides split from the saccharide chain, also known as a glycan. Various monosaccharides are then added to the glycan by means of enzymatic reactions, in the Golgi apparatus. These complex and seemingly unregulated reactions give rise to heterogeneity in a protein's glycan distribution at both macroscopic and microscopic level. Macro-heterogeneity refers to the presence or absence of a glycan on the protein, while micro-heterogeneity refers to the monosaccharide composition of the glycan structure. Specific characteristics of the glycans are known to result in certain pharmacokinetic effects. For instance, terminal galactosylation has been found to impact CDC activity, core fucosylation has been found to impact ADCC activity and terminal sialylation has been found to affect inflammatory properties as well as ADCC activity. In addition, terminal mannose and terminal n-acetylglucosamine (GlcNAc) can reduce serum half-life. Even low abundances of specific glycans have been shown to produce significant pharmacokinetic effects. The final glycan structure formed is known to be dependent on reaction site accessibility, enzyme availability in the host cell, availability of intracellular nucleotide sugar donors, and the transit time in the Golgi apparatus. These internal cellular conditions, which have the most effect on glycosylation, are impossible to control online directly with current technology. However, they can be indirectly manipulated through culture conditions such as bioreactor process variables and media formulation [5].

Factors affecting Glycosylation

Some of the factors that affect glycosylation are:

- Choice of cell line
- Bioreactor operating conditions such as pH, temperature, hydrodynamic stress, and dissolved oxygen
- Nutrient conditions

Challenges faced for online control of glycosylation

A major issue while fulfilling quality assurance and effectiveness, is in the development of a manufacturing process that can generate Mab products having consistent amount of required glycan distribution in each batch. Existing technology can only determine glycosylation patterns after production, resulting in the loss of an entire batch if quality benchmarks do not meet regulations. Online control cannot be implemented because:

- Online measurements for glycosylation are not available.
- Obtaining a comprehensive understanding of glycosylation is challenging due to the complexity of the intracellular process that can vary with cell line, product, and seemingly robust operating conditions.
- Comprehensive control paradigms specific to glycosylation control objectives do not exist.

Modelers' efforts in developing appropriate mathematical representations of underlying biological phenomenon are hindered by complicated glycosylation reaction networks. Several models based on mechanistic and empirical approaches have been developed to overcome these challenges. These predictive models help lay the framework to understand the diversity in glycosylation reaction networks, but a model is still required that will connect extracellular conditions to changes in intracellular conditions and provide real-time predictions of the monoclonal antibody glycan distributions.The following chapters present a modeling framework based entirely on data driven approaches that provides real-time predictions of glycosylation in monoclonal antibodies. The framework rests on three pillars: static approach, dynamic approach and combined approach, all of which have been discussed in the subsequent chapters. The three approaches together present an efficient way of continuously predicting the glycan distribution in monoclonal antibodies. The model is independent of cell lines, bioreactor conditions and the addition of supplements.

Static Approach

The development of a model that uses deep learning to predict the final relative glycan distribution of an antibody has been discussed here. The framework also known as the "Static approach" uses data driven techniques instead of first principles based methods to predict the relative glycan distribution in Monoclonal Antibodies. The aim of this thesis is not to understand the underlying physics behind glycosylation in monoclonal antibodies but to develop an efficient model that provides accurate and instantaneous predictions of the glycan distribution from experimental data.

The model building approach consists of the following parts:

- Data Collection
- Data Pre-processing
- Data Visualization and Analysis
- Deep Learning Model
- Model Prediction
- Model Evaluation

Data Collection

The Data Collection part of the model involved running a series of experiments to collect the data required to build the model. An autoclavable bioreactor system provided by Applikon was used to conduct the experiments. An industrial telecommunication standard, Open Platform Communication (OPC) was used to communicate with the bioreactors. It was chosen primarily because OPC is platform independent and any OPC enabled application can freely communicate with any OPC enabled source without any concern for the vendor it came from. Additionally, an OPC enabled application can communicate with as many sources as needed, or in other words, there are no limitations to the number of connections that can be made between data sources and receivers. The entire experimental setup consisted of the following parts:

- A bioreactor with the appropriate auxiliaries - stirrer assembly, sensors, an aeration assembly etc. for process control.
- A my-Control bio-controller for measurement and control of process variables (like pH, temperature, dO2, level and stirrer speed) with corresponding actuators in order to keep process conditions at set point.
- A host PC that is used as a Human Machine Interface (HMI) used to provide set points to the controller.
- An Open Platform Communication (OPC) server by Applikon that converts the hardware communication protocol into the OPC protocol.
- An OPC client by MATLAB that connects to the hardware. The OPC client uses the OPC server to get data from or send commands to the hardware.

Open Platform Communications [10] or OPC is a series of standards and specifications that facilitates the exchange of data between an industrial hardware and the human machine interface. OPC has two components, the server and the client. The client makes a request to the server that fulfills the request. The OPC server is a software program that converts hardware communication protocol into OPC protocol and vice versa [11]. It also responds to requests from one or more OPC clients to provide data. An OPC client is a software that translates a given applications communication request into an equivalent OPC request and sends it to the OPC server for further processing. It also translates OPC Data coming from the server back to the applications native language. The most common type of data transferred through OPC are

- Real Time data
- Historical data
- Alarm & Event data

The three specifications corresponding to the categories are:

- OPC Data Access (OPC DA)

- OPC Historical Data Access (OPC HAD)
- OPC Alarms and Events (OPC A&E)

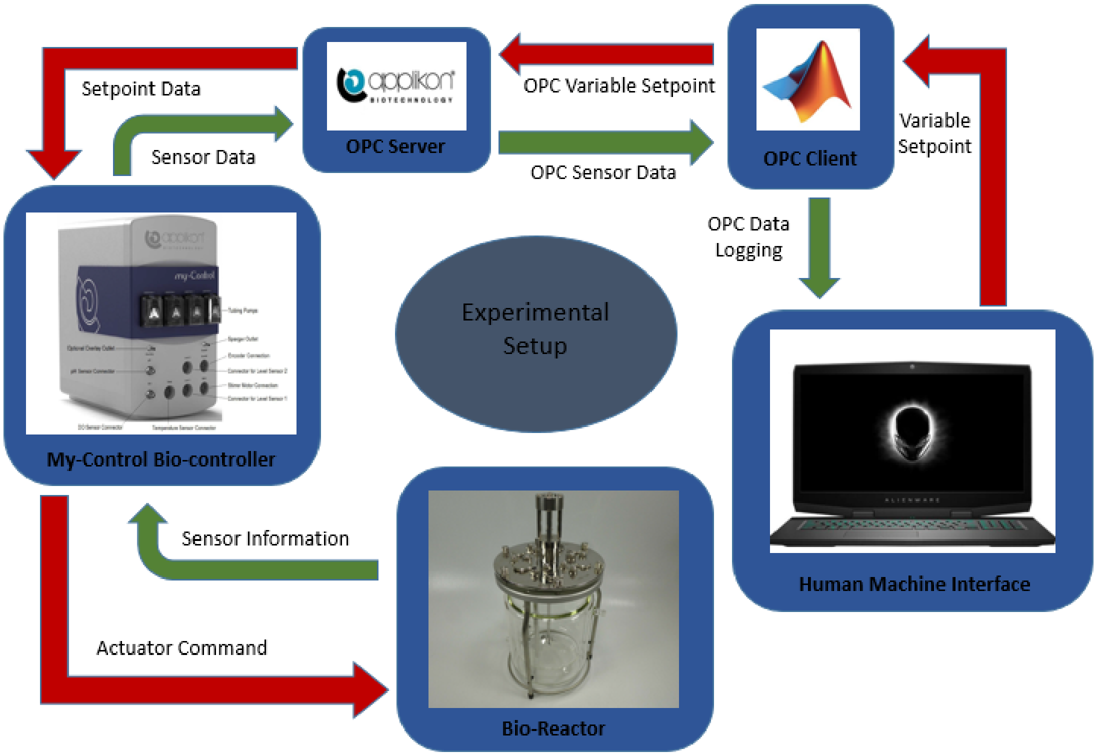


Figure 5: Experimental Setup and Flow of Information

The flow of information in the experimental setup has been shown in Figure 5. The experiments are performed in the bioreactor provided by Applikon that has its own bio-controller. The bio-controller has access to all the information from the sensors of the bioreactor and it controls the actuators that are responsible to keep the process variables within range. It also has inbuilt PID controllers that provide actuator commands corresponding to the set point of a particular variable. The information received from the sensors of the bioreactor is converted from the bioreactor communication protocol to the OPC protocol by Applikon's OPC server. The OPC server responds to the data access request sent by the OPC client software installed in a host PC and provides it with the requested data. The OPC client converts the sensor information from the OPC protocol to the native applications communication protocol and displays it to the user through the Human Machine Interface. The user logs the sensor information and may decide to change the set point depending on the current value of the variable. In that case, the variable set point value is converted back to the OPC protocol by the client, which makes a write data request to the server. The server receives that request and converts it to the bio-controller communication protocol. The bio-controller receives the set point information and feeds it to its PID controller that in turn provides actuator commands to keep the variable values within range.

Each experiment was run for 7 days and the data logging process continued throughout the entire time span of the experiment. The following information was collected from the experiment:

- Glucose Concentration Profile: Glucose is one of the key nutrients required for cell growth in mammalian cell cultures.
- Glutamine Concentration Profile: Glutamine is the other key nutrient required for cell growth.
- Lactate Concentration Profile: Lactate is formed as a by-product during cellular uptake of glucose. It is known to inhibit cell growth in mammalian cells. However, some cells are known to consume lactate under low glucose conditions.
- Ammonia Concentration Profile: Ammonia is formed as a by-product during cellular uptake of glutamine. It is also known to inhibit cell growth.
- Viable Cell Density Profile: It is the number of living cells in a cell culture. It depends on the cell growth rate and death rate
- Total Cell Density Profile: It is the total number of cells in a cell culture. It depends on the number of viable cells.
- Antibody Concentration Profile: Measure of the protein productivity
- pH Profile: The pH profile was kept in the range between 7.00 and 7.20 since it has been observed that the highest specific growth rate and maximum viable cell concentration is obtained in that range.
- Dissolved Oxygen profile: Like pH, the dissolved oxygen percentage also has an impact on the final glycan distribution. It was held constant at a certain level to maximize productivity.
- Temperature profile: The temperature was held constant at 37°C

- Stirrer speed profile: The stirrer speed was another parameter that was kept constant at a certain speed.
- Final relative glycan distribution: The 9 observed glycans that represent the glycosylation profile in monoclonal antibody are as follows:
    - A1:
    - FA1
    - A2
    - FA2
    - A2G1
    - FA2G1
    - FA2G1'
    - A2G2
    - FA2G2

  The final relative distribution of these 9 glycans was collected at the end of the experiment. These glycans can also be grouped together based on the presence of galactose or Fucose group. Successfully predicting the grouped percentage can also help to determine the final product quality of an antibody. The glycans above can be divided into the following 3 groups:

    - G0 + G0F
    - G1 + G1F
    - G2 + G2F

  The relative percentages of each of these three groups in the final glycan distribution were calculated. The Fucose percentage among all the glycans is also an important descriptor of the final product quality of an antibody. Therefore, the percentage Fucose level among all these glycans was also measured.

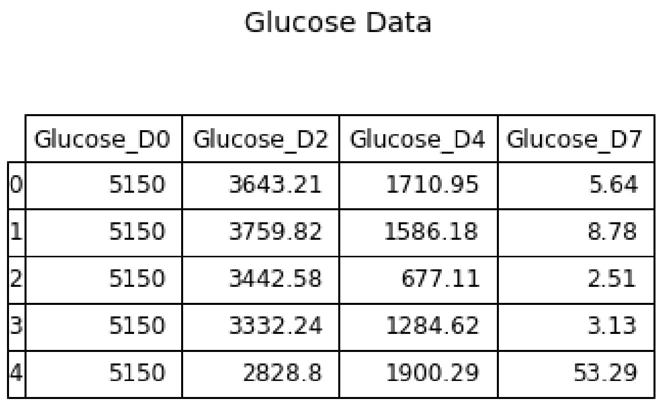
The final model presented below has the capability of predicting all of these different outputs.

Data Preprocessing

The Data Preprocessing step involved selecting the variables that will have the most influence on the final glycan distribution. In congruence with previous literature reviews, the following variables were found to have the most impact on the glycan distribution:

- Glucose Concentration Profile
- Glutamine Concentration Profile
- Lactate Concentration Profile
- Ammonia Concentration Profile
- Viable Cell Density Profile
- Total Cell Density Profile
- Antibody Concentration Profile

These variables were then divided into 4 time steps (Day-0, Day-2, Day-4, Day-7) that covered the entire timespan of the experiment and using these 7 variables each having 4 time steps, a 28 dimensional feature space was created. Figure 6 shows Glucose concentration profile divided into 4 time steps for the first five experimental data.



Glucose Data

	Glucose_D0	Glucose_D2	Glucose_D4	Glucose_D7
0	5150	3643.21	1710.95	5.64
1	5150	3759.82	1586.18	8.78
2	5150	3442.58	677.11	2.51
3	5150	3332.24	1284.62	3.13
4	5150	2828.8	1900.29	53.29

Figure 6: Glucose Concentration Profile divided into 4 time steps

The multidimensional feature space was highly correlated since it contained time series data of the variables and for each variable, the future time steps were dependent on the previous time steps. Moreover, the variables were also dependent on each other. For accurate predictions, it is extremely desirable to have independent and uncorrelated variables in the feature space. Else, the model might give inconsistent predictions. The Data Visualization and Analysis part of the model handled this aspect.

Data Visualization and Analysis

In the Data Visualization and Analysis part of the model, the correlation coefficient of the independent variables were found using Pearson's Correlation and Spearman's correlation to check for both linear and monotonous interactions between the variables. Pearson's Correlation, is the measure of the linear correlation between two variables. The heat map generated by the Pearson correlation coefficient calculation is given in Figure 7. It showed that there were more than 182 linear interactions that were either positively (r score greater than 0.5) or negatively (r score less than -0.5) correlated.

Similarly, Spearman's correlation, which determines the strength and direction of any monotonic relation between two variables, showed that there were 118 monotonous interactions between all the variables. The heat map for spearman's correlation coefficient is given in Figure 8. This proved that most of the variables were correlated either linearly or non-linearly. Hence, to get accurate prediction results from the model, we had to construct a feature space that consisted of independent and uncorrelated features. To do that, a statistical technique called Principal Component Analysis was used. Principal Component Analysis is a statistical evaluation that is used to convert a set of possibly correlated variables into a set of linearly uncorrelated variables called the principal components. It also helps to select the components that account for maximum variance in the dataset. This reduces the dimension of the feature space and significantly increases computational efficiency. With the help of Principal Component Analysis, our initial 28 dimensional feature space was reduced to six components that explained nearly 95% of the variance in the dataset.

This preprocessed data in the newly constructed feature space was then used to train the deep learning model.

Deep Learning Model

The deep learning model is a fully connected neural network with six inputs, multiple hidden layers and variable outputs. The number of hidden layers was decided by optimizing the hyperparameters of the model and the output size was dependent on the type of glycan prediction.

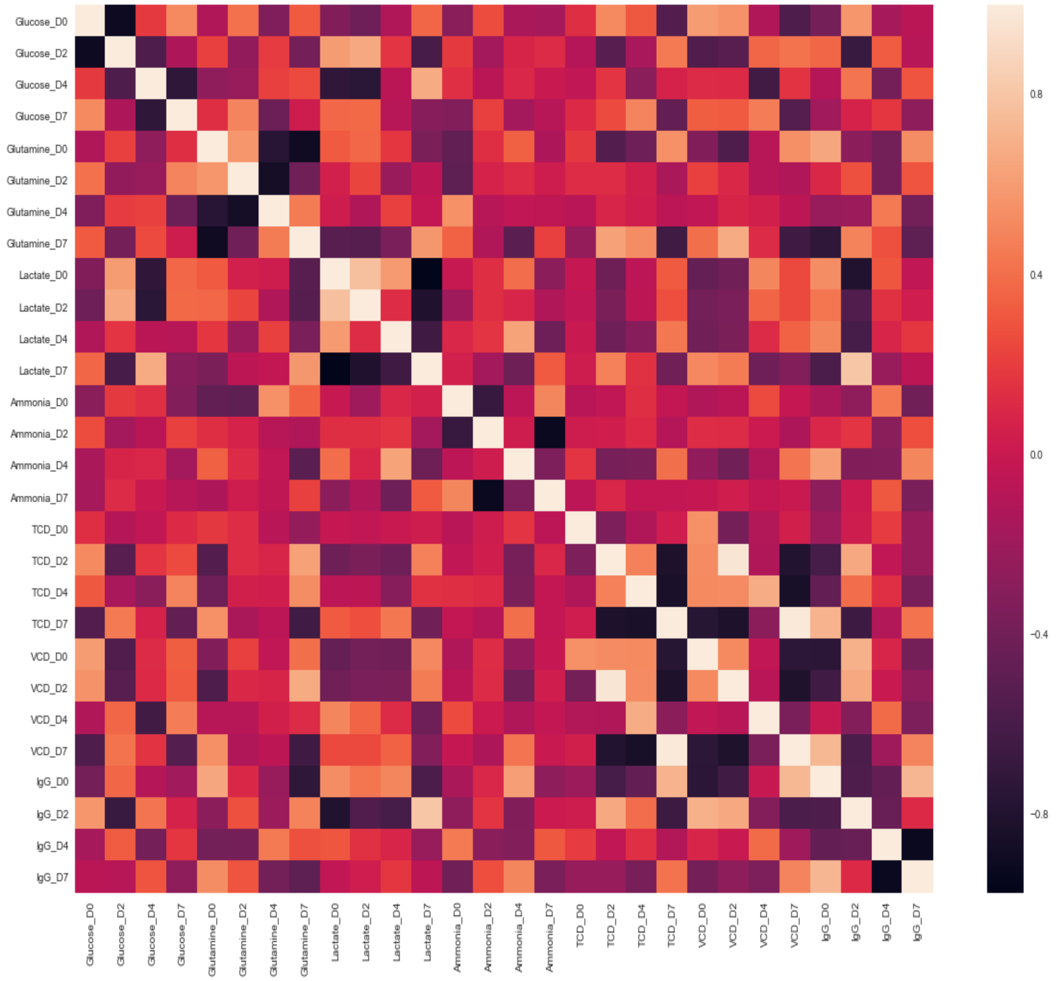


Figure 7: Pearson's Correlation Heat Map

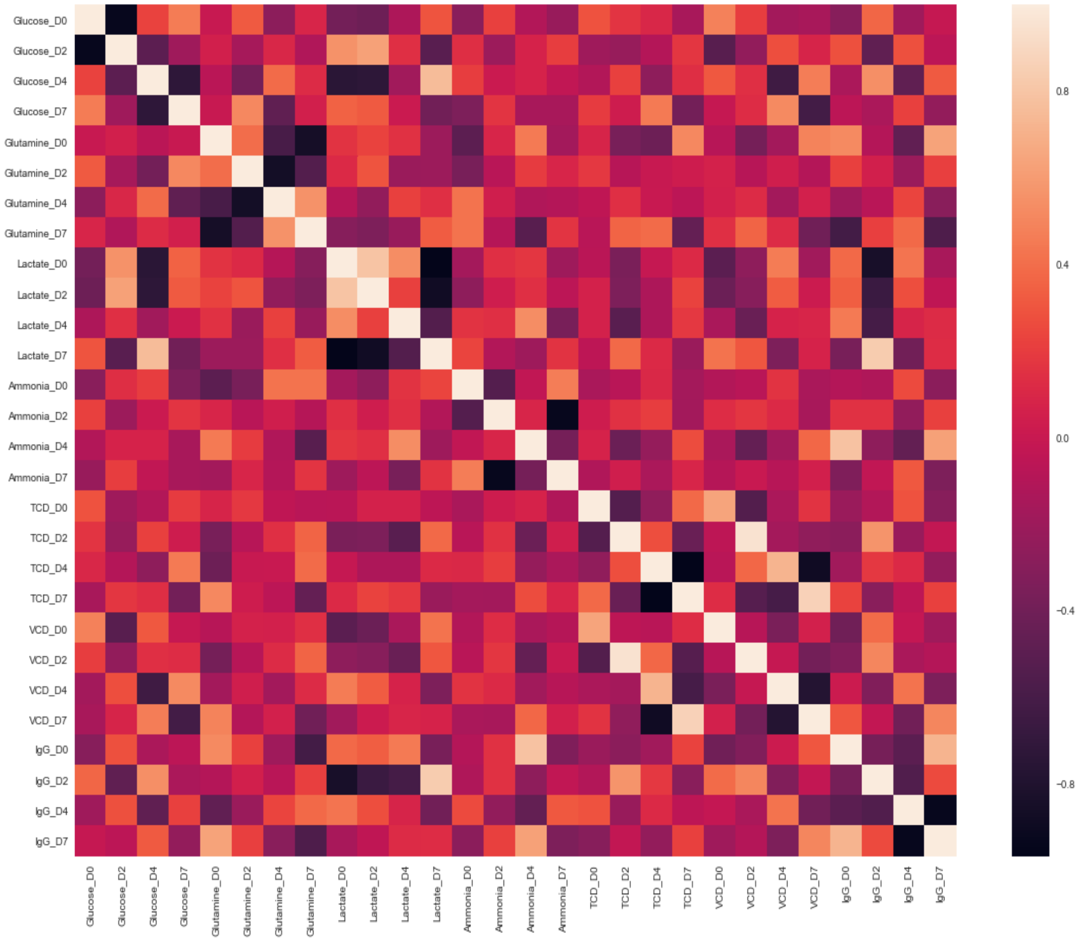


Figure 8: Spearman's Correlation Heat Map

A neural network is a multi-layer network of nodes that is used to make predictions by "learning" from examples or data. Figure 9 below shows the multiple layers of a fully connected neural network [12]. Starting from the left, the first layer is the input layer, the second and third are the hidden layers and the final layer is called the output layer. There can be multiple hidden layers depending on the model architecture. The arrows connecting the layers represent the flow of information through the interconnected neurons from the first to the last layer. A forward pass operation helps a neural network make a prediction while a backward pass operation helps it to minimize the error in prediction. The forward pass is a series of linear matrix-vector operation

(Matrix of weights and vector of input layers) followed by a nonlinear operation on the nodes by an activation function. Mathematically, it can be represented as:

$$h_1 = f(W_x X) - (1)$$

$$h_n = f(W_{hn} h_{n-1}) - (2)$$

$$y = g(h_n) - (3)$$

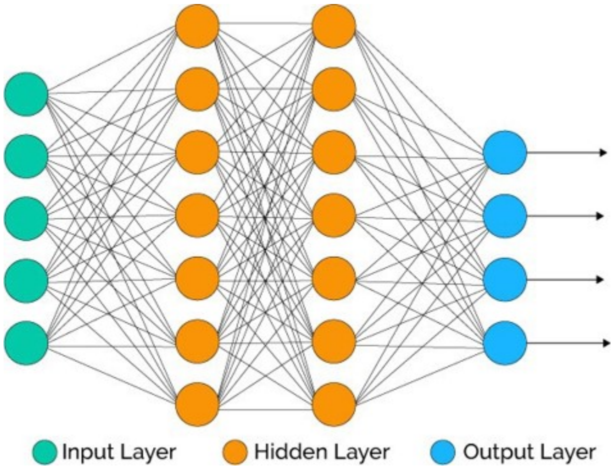
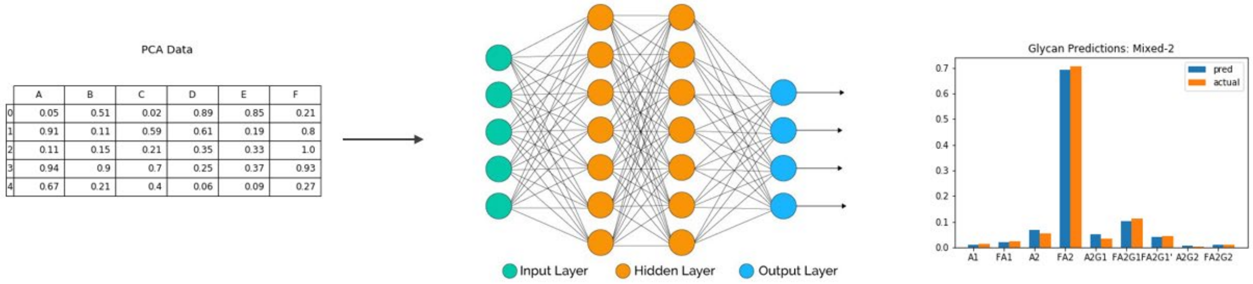


Figure 9: A Fully Connected Neural Network

In Equation (1), h_1 is the value of the first hidden layer vector. It is obtained by performing a matrix vector multiplication of the Weight matrix, $W_{x,}$ between the input layer, X, and the hidden layer, h_1, and then passing it through a nonlinear function denoted by f. In Equation (2), h_n is the value of the nth hidden layer vector. It is obtained by performing a matrix vector multiplication of the weight matrix, $W_{hn,}$ between the hidden layer, h_n, and the previous hidden layer, h_{n-1}, and then passing it through the nonlinear function f. In Equation (3), the output prediction of the forward pass y is obtained by passing the final hidden layer vector through a function g. With the prediction obtained from the forward pass and the original value of the output, a loss function is calculated. The backward pass calculates the gradient of the loss function with respect to the weight matrices and keeps on updating it, till a certain error tolerance or maximum number of iterations is reached, using an optimization algorithm. This Trained model can then be used to make predictions on unknown dataset. The number of hidden layers, learning rate of the optimization algorithm, activation function, and loss function are all hyperparameters of the model that governs the generalization capability of the model. They are chosen using a technique called hyperparameter optimization where the performance of the model is evaluated on a separate held out dataset that is not used to train the model.

The fully connected neural network of our model has six input nodes that takes in the six components obtained from principal component analysis of the data. The number of output nodes depends on the type of relative glycan distribution prediction that is needed. The relative glycan distribution prediction can be of three types:

- 9 glycan output
- 3 grouped glycan output
- Percentage Fucose output


PCA Data

	A	B	C	D	E	F
0	0.05	0.51	0.02	0.89	0.85	0.21
1	0.91	0.11	0.59	0.61	0.19	0.8
2	0.11	0.15	0.21	0.35	0.33	1.0
3	0.94	0.9	0.7	0.25	0.37	0.93
4	0.67	0.21	0.4	0.06	0.09	0.27

Figure 10: Deep Learning Model Visualization

Depending on the type of glycan distribution, the output layer will consist of 9, 3 or 1 node. The preprocessed experimental data is then divided into the training set and the validation set. The model was trained using the training set and the validation set was used to obtain the optimal set of hyperparameters and to evaluate the model. Python's scikit-learn module [13], which provides access to several efficient tools for data analysis and machine learning, was used to preprocess the data, train the neural network and choose the best set of hyperparameters. The visual representation of the neural network model along with the inputs and the outputs is presented in Figure 10. It shows how the data obtained from Principal Component Analysis is used as an input to the fully connected neural network and it predicts the 9-glycan relative distribution in monoclonal antibodies.

Multiple iterations were carried out to select the best possible set of hyperparameters using scikit learn. 100 different neural networks were trained on each set of hidden layers and hidden units by varying the random state parameter in scikit learn. Then the average scores of those 100 neural networks on the validation dataset were used to determine the best possible hidden layer and hidden unit combination. The result of the iterations is given in Figure 11. It is clear from the given heat map that the best possible combination of hidden units and hidden layers is 3 and 4 respectively. Similarly, other hyperparameters like learning rate, loss function, number of training epochs were also optimized using the same technique. The final model architecture after hyperparameter optimization is as follows:

- Input Layer Shape: 1x6
- Output Layer Shape: 1x9 or 1x3 or 1x1
- Hidden Units: 3
- Hidden Layers: 4
- Activation Function: “Logistic”
- Learning rate: 0.001
- Regularization Penalty: L2
- Regularization parameter: 0.0001
- Number of iterations: 300
- Error Tolerance: 1e^-5

Model Prediction

The model obtained after hyperparameter optimization was then used to make predictions. Figure 12 and 13 below show the model predictions on two test cases where the output is a 9 glycan relative distribution. These two test cases were chosen to show that the model is independent of two different cell lines. Figure 12 shows the relative glycan distribution prediction for CDOptiCHO cell line where the glycan FA2 has a much higher relative percentage compared to the other glycans while Figure 13 shows the prediction for AMBIC cell line where even though the glycan FA2 has a higher relative percentage than other glycans, it is still much less than the FA2 in CDOptiCHO cell line. The blue bars in both the graphs represent the predicted value while the green bars represent the actual value. From the two figures, it is evident that the model has learnt to predict both the cell lines with similar accuracy and hence it proves the claim that this model is independent of CDOptiCHO and AMBIC cell lines.

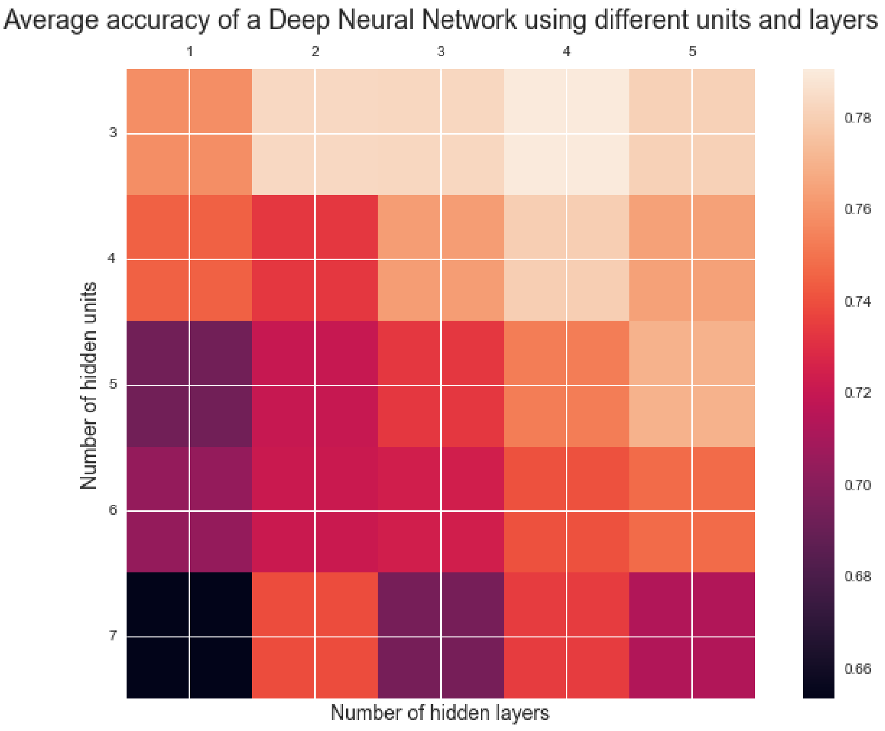


Figure 11: Hyperparameter Optimization

Model Evaluation

The model was then evaluated on all the test cases using 3 different metrics of model evaluation:

- R-squared value
- Mean Squared Error
- Mean Absolute Error

A multivariate linear regression model was also trained using the same training dataset and evaluated on the same validation set. The results of both these models are presented in Figure

14. It shows that the Deep learning model has a better model metric score than the multivariate Linear Regression model in all three cases.

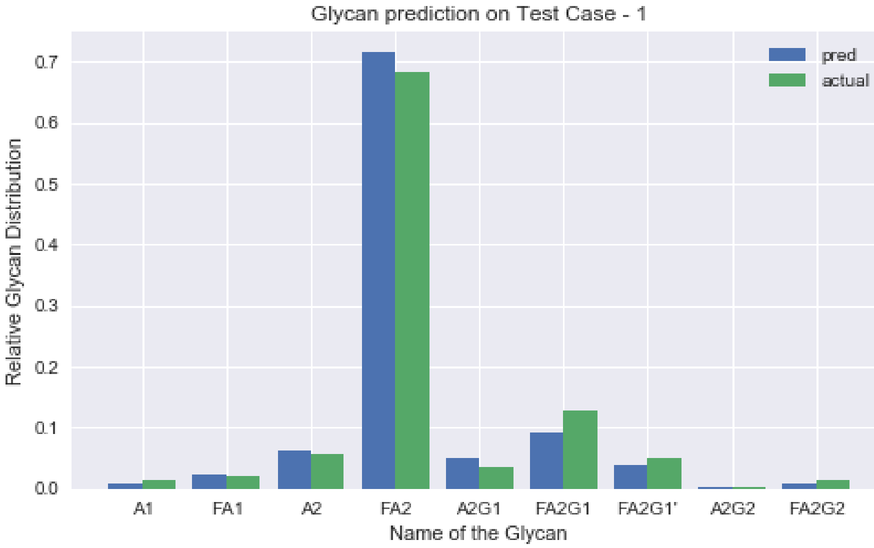


Figure 12: Glycan Prediction on test case 1

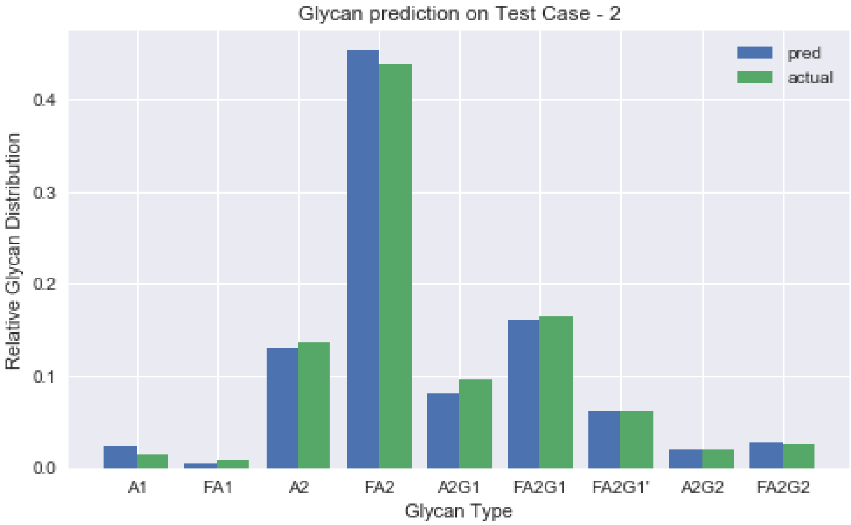
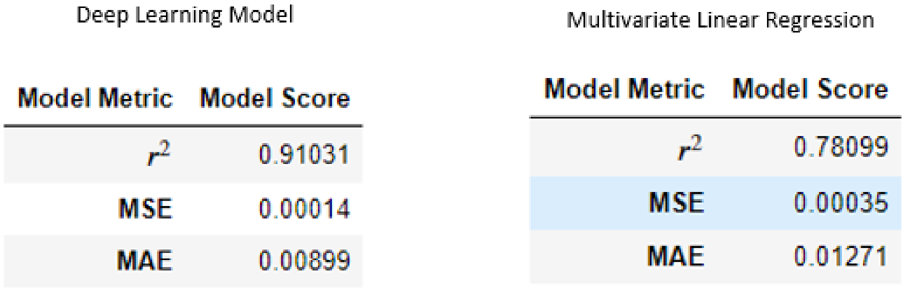


Figure 13: Glycan Prediction on test case 2

The static approach discussed above can predict the final relative glycan distribution in monoclonal antibodies accurately and efficiently given the media concentration data, total and viable cell densities and the antibody concentration profile throughout the time span of the experiment. However, it still fails to predict the glycan distribution continuously as the experiment progresses. The dynamic approach helps to eliminate this shortcoming of the static approach. It will be discussed in the next section.



Deep Learning Model

Model Metric	Model Score
r^2	0.91031
MSE	0.00014
MAE	0.00899

Multivariate Linear Regression

Model Metric	Model Score
r^2	0.78099
MSE	0.00035
MAE	0.01271

Figure 14: Deep Learning Model vs Linear Regression Model

Dynamic Approach

For an experiment that runs for N time steps, the dynamic approach presents a framework that predicts the concentration profile of the antibody for the final N - x time steps, using data from the first x time steps. While the static approach achieved high accuracy in relative glycan prediction, it required data throughout the entire timespan of the experiment to make that prediction. The aim of this thesis is to make real-time predictions of the glycan distribution. The first step towards attaining that goal was to predict the concentration profiles of the nutrients, antibody, total cell density and viable cell density for future time steps of the experiment given experimental data until the current time step. Using the dynamic approach, we were able to achieve that goal.

The approach rests on a deep learning model called recurrent neural network (RNN) that is a collection of neural networks where the output of the hidden state of the previous neural network if fed as input to predict the next state of the current neural network. In a traditional neural network, as shown above in the static approach section, all the inputs and outputs are independent of each other. However, in case of time series prediction, the input, which is usually the information until the current time steps, is related to the output, which is usually the prediction of the next time step value. Hence, there is a requirement to remember the previous time step input in order to predict the next time step value. A recurrent neural network has the capability of capturing this information by "sharing the states" at every time step [14]. The most important feature of a recurrent neural network is its hidden state, which "remembers" information about the previous states. The inner workings of a recurrent neural network is shown in Figure 15. It shows an RNN that uses not only the input at time step t denoted by X_t but also the hidden state of the network at time step t-1 denoted by h_{t-1} to predict the hidden state at time step t denoted by h_t. The hidden state of an RNN is the value of the hidden units after performing the matrix vector operation on the weight matrix and input vector and passing it through a non-linear function called the activation function.

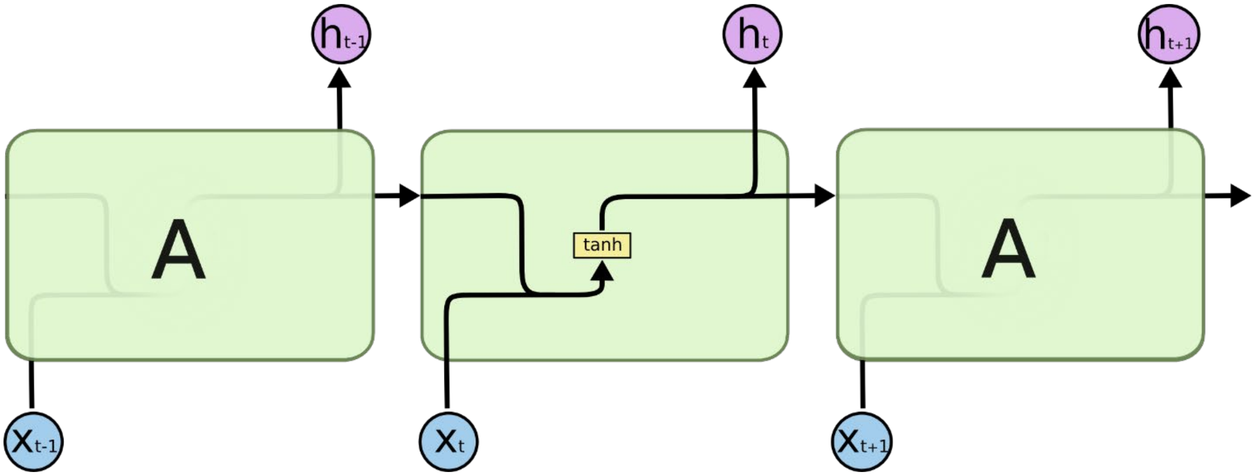


Figure 15: RNN visualization

Mathematically, the hidden state calculation of an RNN is like a fully connected neural network but with an additional weight matrix W_{hh}.

$$h_t = f(W_{hh}h_{t-1} + W_{xh}X_t)$$

In this case, the hidden state at time step t h_t is calculated by performing 2 matrix-vector multiplication operations, summing the result and passing it through a nonlinear function. Here, W_{hh} is the weight matrix between the hidden states at two different time steps, W_{xh} is the weight matrix between the first hidden layer and input layer, h_{t-1} is the hidden state vector at the previous time step and X_t in the input layer vector. Hence, the current state is not only a function of the input at time step t but also a function of the previous state at time step t-1. This way the RNN remembers the previous output and uses it to make predictions. While theoretically, RNNs can remember long-term dependencies in time steps, practically, they cannot. This problem is referred to as the vanishing gradient problem where more and more importance is given to the nearer time steps and thus an RNN tends to forget time steps that are far away from the current time step. A solution to this problem is using an LSTM cellblock instead of a simple activation function that is normally used in an RNN. LSTMs or Long Short Term Memory networks take care of the vanishing gradient problem by remembering information for longer periods [15]. An LSTM cellblock is shown below in Figure 16. They have the same structure as an RNN but instead of having a single activation function, there are 4 different functions interacting in a very special way. The details about the internal structure of an LSTM and how they solve the vanishing gradient problem is beyond the scope of this thesis.

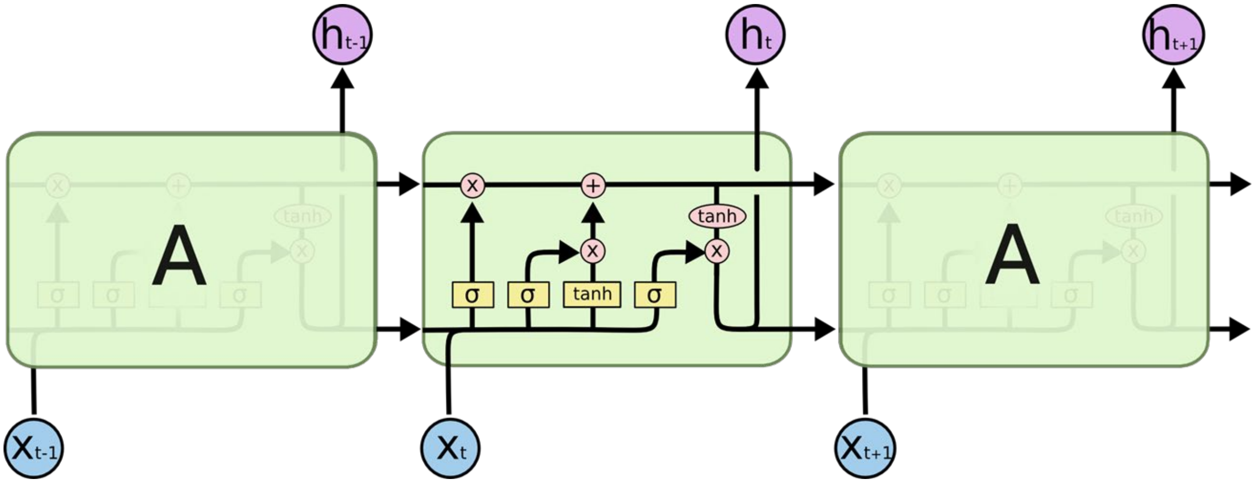


Figure 16: LSTM cell block visualization

The dynamic model presented in this thesis uses Recurrent Neural Network with LSTM cells. The available experimental dataset as shown above was a combination of time series data and cross-sectional data. Since it was multivariate, simple univariate time series analysis like AR, ARMAX will not be the best choice and since it was pooled, multivariate analysis methods like VARMAX will not work either. Therefore, we have chosen a Recurrent Neural Network with LSTM cells as the primary framework of the dynamic approach. The variables chosen to build the dynamic model were:

- Glucose Concentration Profile till time step i
- Glutamine Concentration Profile till time step i
- Lactate Concentration Profile till time step i
- Ammonia Concentration Profile till time step i
- Viable Cell Density Profile till time step i
- Total Cell Density Profile till time step i
- Antibody Concentration Profile till time step i

The objective of the model is to predict the entire antibody concentration profile continuously, i.e., given all the information of the above variables until a particular time step, the dynamic approach predicts the antibody concentration profile from time step i +1 to N, N being the last time step of the experiment. It is represented visually in Figure 17. The steps to build the model were as follows:

- Data Preparation
- LSTM Network training and prediction
- Model Evaluation

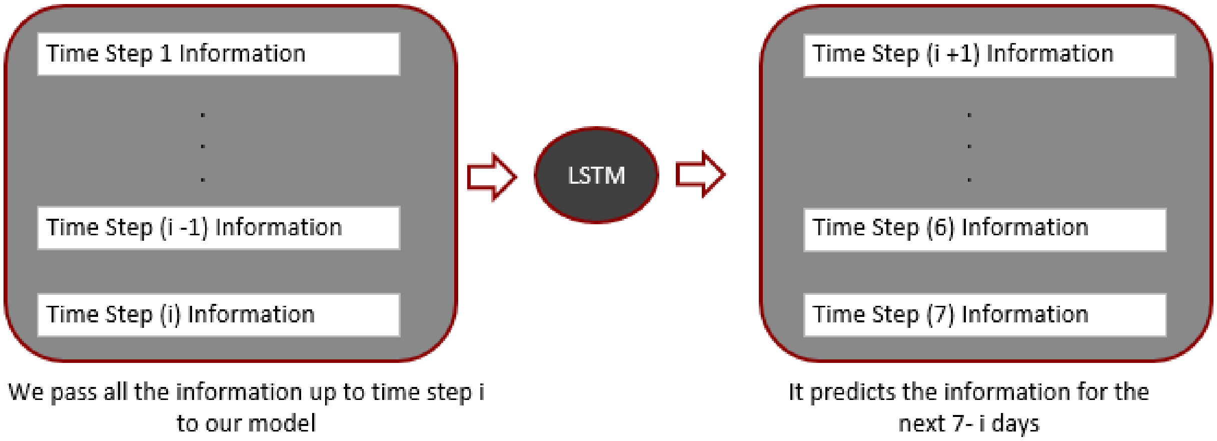

Time Step 1 Information
.
.
.
Time Step (i -1) Information
Time Step (i) Information
We pass all the information up to time step i to our model
LSTM
Time Step (i +1) Information
.
.
.
Time Step (6) Information
Time Step (7) Information
It predicts the information for the next 7- i days

Figure 17: Workings of the Dynamic Model

Data Preparation

The Data Preparation step involved creating a 7-dimensional feature space that represented the 7 variables used for prediction. The number of time steps used for each variable depended on the availability of information. At a particular time step, a tensor of shape (m*n*7) was created where m represents the number of experimental examples, n the number of available time steps and 7 the number of variables. This data was then used to train the dynamic model.

LSTM Network training and prediction

An Open Source Machine Learning Platform, TensorFlow [16] was used to create and train the model. It provides access to the Keras API that has built in functions required to design the model. The prepared data was then used to train the model for multiple epochs. The data consisted of the 7 variables listed above; each having 4 time steps. It was split into training and testing dataset. The model was first trained on only the first two time steps and the trained model was used to predict the next two time steps. The entire concentration profile of the antibody along with the prediction of the last two time steps and the actual values for the first 3 test cases is shown in Figure 18. Here, information of all the variables until Day 2 or the second time step was fed to the dynamic model and it predicted the last two time steps, Day 4 and Day 7. Then the model was trained on the first three time steps and made to predict the final time step. The concentration profile of the antibody along with the predicted and actual values of the final time step for the first 3 test cases is shown in Figure 19. Here, data collected for all the variables until Day 3 or the third time step was fed to the dynamic model and it predicted the last time step, Day 7. The model was thus able to make continuous predictions of the antibody concentration profile throughout the time span of the experiment. The final model architecture of the dynamic model is as follows:

- Number of hidden units in LSTM cell: 4
- Number of hidden LSTM layers: 1
- Learning Rate: 0.001
- Number of training epochs: 200
- Error Tolerance 1e-7

- Loss Function: Mean Squared Error
- Optimizer: Adam

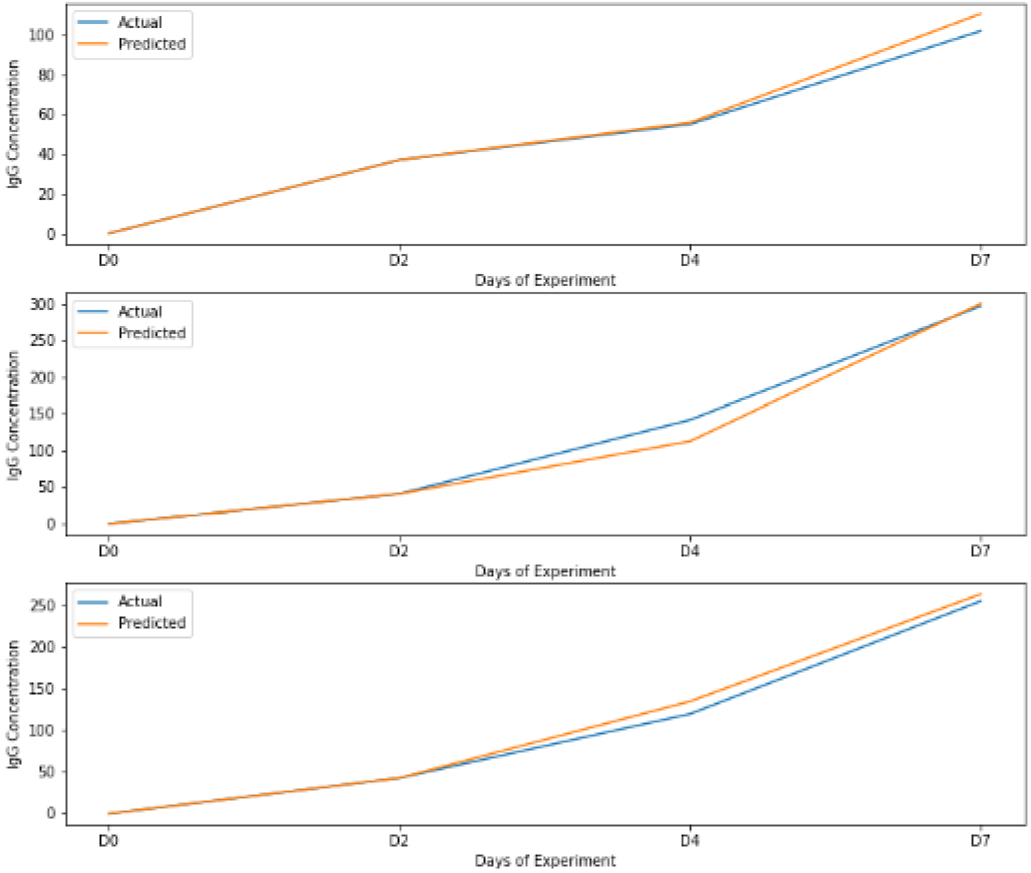


Figure 18: Two-step prediction of antibody concentration profile

Model Evaluation

In the model evaluation step, the mean squared error between the predicted and actual values were calculated to measure the model performance on unseen test cases. The One-step prediction model gave a mean squared error of 0.004734 and the Two-step prediction model gave a mean squared error of 0.005796 on test cases after training for 200 epochs.

This model was the second step towards achieving our goal to make real time predictions of the glycan profile in monoclonal antibodies. The continuous predictions of antibody concentration profile obtained from this model can be used later in the combined approach, which is discussed in the next section of this thesis.

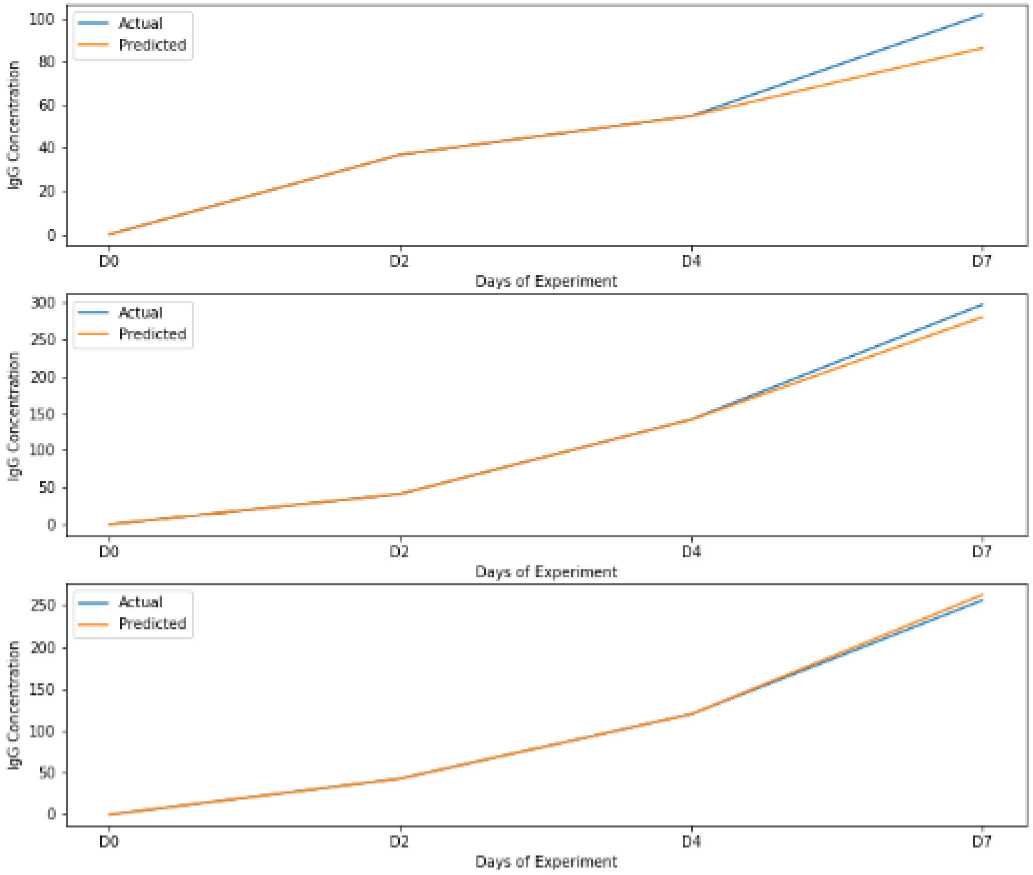


Figure 19: One-step prediction of antibody concentration profile

Combined Approach

The static approach introduced a way to calculate the relative glycan distribution if information gathered from the required variables was available throughout the entire timespan of the experiment. The dynamic approach on the other hand provided a method to predict the antibody concentration profile continuously with information from the variables until a particular time step. The combined approach, as the name suggests, is a union of the static and dynamic approach and it presents a framework to predict the final relative glycan distribution in monoclonal antibodies continuously as the experiment progresses.

A visual representation of the combined approach is shown below in Figure 20. If we had the following information from an experimental run:

- Glucose Concentration Profile till time step i
- Glutamine Concentration Profile till time step i
- Lactate Concentration Profile till time step i
- Ammonia Concentration Profile till time step i
- Viable Cell Density Profile till time step i
- Total Cell Density Profile till time step i
- Antibody Concentration Profile till time step i

We fed that information to the LSTM network based dynamic model that would predict the entire antibody concentration profile for the experiment. We used the predicted information as input to our static model and it predicted the final relative glycan distribution as output. Thus, we were able to achieve real time predictions of glycosylation in monoclonal antibodies.

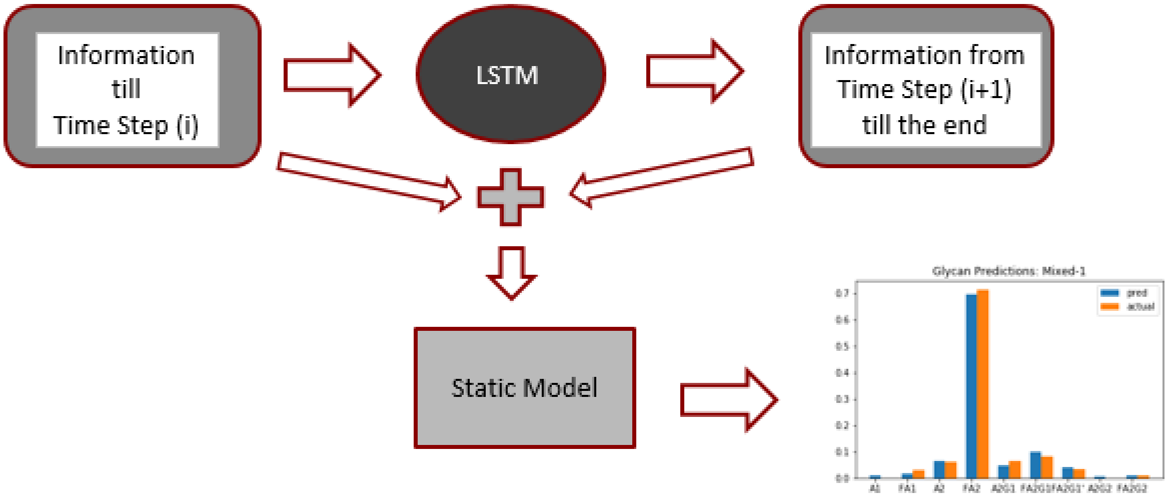

Information till Time Step (i)
LSTM
Information from Time Step (i+1) till the end
Static Model
Glycan Predictions: Mixed-1
pred
actual
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
A1
FA1
A2
FA2
A2G1
FA2G1
FA2G1'
A2G2
FA2G2

Figure 20: Combined Approach Visualization

Modified Static Model

The static model used in the combined approach is slightly different than the one described above in the static approach section of the thesis, which required information about 7 variables. However, the modified static model only requires the Antibody concentration profile as input. The Data Preparation stage of the modified static model involved the following steps:

- Breaking the Antibody Concentration Profile into time steps and creating an input space of shape (m*n) where m is the number of examples and n is the number of time steps
- Normalized the input space using the z-score standardization technique
- Decomposing the normalized data into a linearly uncorrelated and independent feature space using Principal Component Analysis
- Splitting the feature space into training set and testing set

After preparing the data, the model architecture was also altered slightly to increase the model performance. The new model architecture is described below:

- Input Layer Shape: 1x4; It is a vector with 4 units compared to the 6 unit feature space used in the previously described static model
- Output Layer Shape: 1x3 or 1x1; Unlike the original static model which had the capability to predict the 9 glycan relative distribution, this model was used to predict only the 3 glycan grouped distribution and the percentage Fucose level. This was imposed to increase the stability of the model.
- Hidden Units: 3
- Hidden Layers: 2
- Activation Function: “Logistic”
- Learning rate: 0.001
- Regularization Penalty: L2
- Regularization parameter:).0001
- Number of iterations: 300
- Error Tolerance: 1e^-5

Using the modified static model described above and the dynamic model described in the dynamic approach section of this thesis, the combined model was able to make real time predictions of the glycan distribution in monoclonal antibodies. The Combined Approach had the following stages:

- Data Processing
- Static Model Training

- Dynamic Model Training
- Model Prediction
- Model Evaluation

Data Processing

The Data Processing stage had the following steps:

- Processed data for the 7 variables, Glucose Concentration, Glutamine Concentration, Lactate Concentration, Ammonia Concentration, Total Cell Density, Viable Cell Density, and Antibody Concentration continuously
- Selected the 4 major time steps at Day 0, Day 2, Day 4 and Day 7 and logged them into memory using OPC
- Prepared the data for the static and dynamic model
- Split the data into training and test dataset

Static Model Training

In the Static Model Training step, the modified static model was trained using the antibody concentration profile only.

Dynamic Model Training

The Dynamic Model was first trained using the data for the 7 variables for the first two time steps, Day 0 and Day 2 only. Then it was trained again using the data for the 7 variables for the first 3 time steps, Day 0, Day 2, and Day 4

Model Prediction

During the Model Prediction stage, the trained dynamic model was used to predict multiple time steps for each test case. First, it was used to predict the antibody concentration profile given the first two time steps, Day 0 and Day 2. Then, it was again used to predict the antibody concentration profile given the first 3 time steps, Day 0, Day 2 and Day 4. Figure 21 below shows the dynamic model output for 3 test cases. Here the model predicts the antibody concentration profile given the first 3 time steps Day 0, Day 2 and Day 4. Then, the modified static model trained previously was used to predict final relative grouped glycan distribution using the predicted antibody concentration profile given by the dynamic model. Figure 22 below shows the grouped relative glycan distribution predicted by the modified static model.

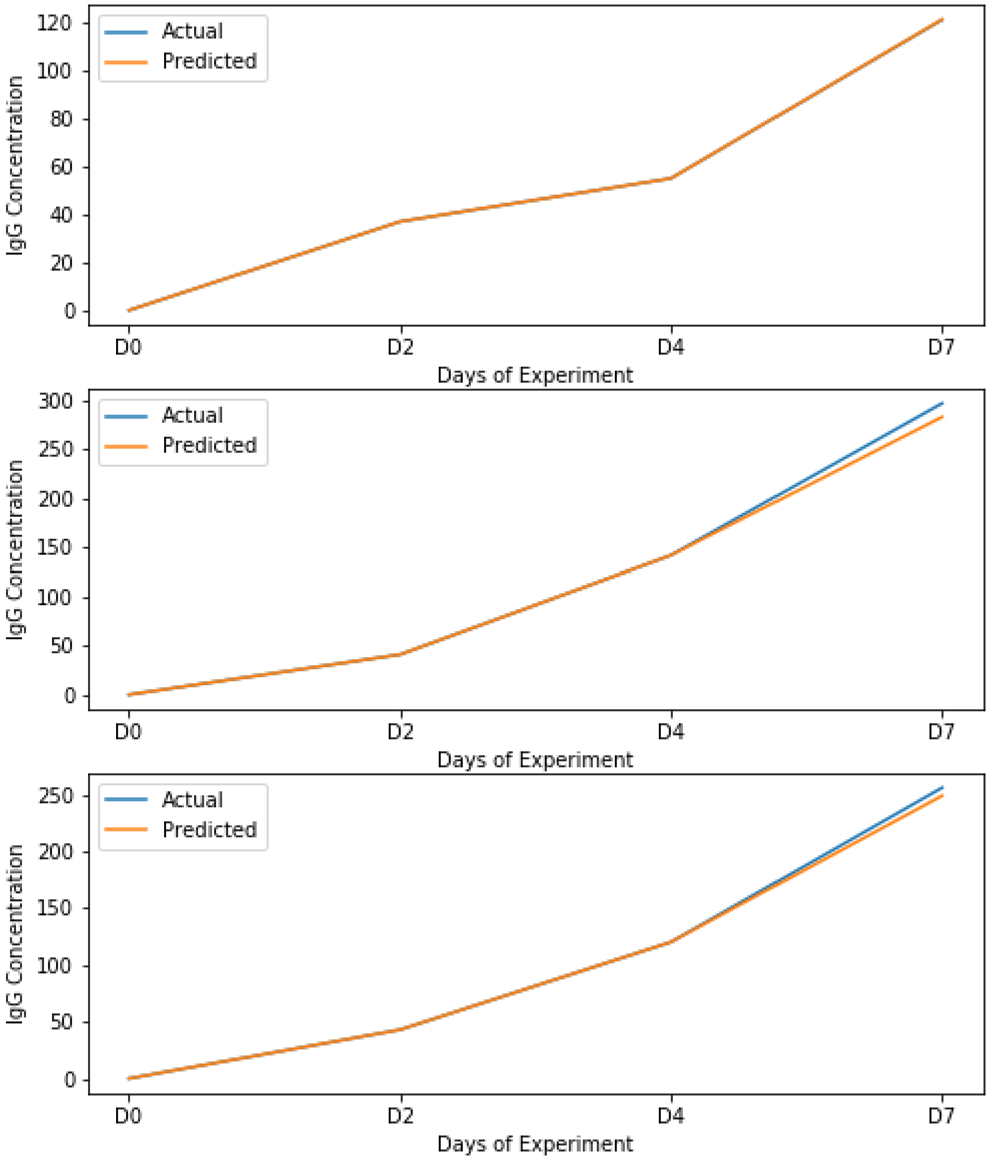


Figure 21: Antibody Concentration Profile prediction by the dynamic model

Model Evaluation

The mean squared error of the combined approach on the test cases for the grouped relative glycan distribution output was 0.00219. Using the static and dynamic approach together helped us to make real time predictions of the final relative grouped glycan distribution accurately and efficiently. This model can be used for multiple purposes as discussed in the conclusion section of this thesis.

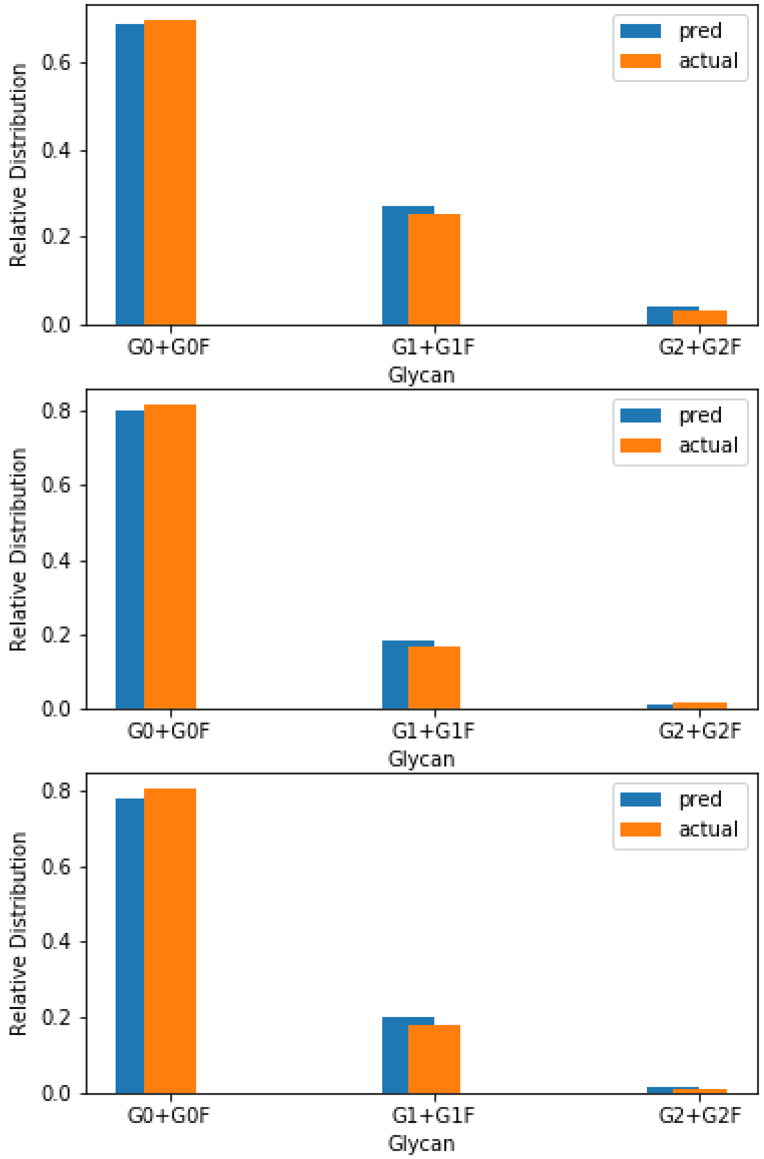


Figure 22: Final relative grouped glycan distribution by the modified static model

Conclusion

The proposed framework presents an efficient way to make real time predictions of the final relative glycan distribution of monoclonal antibodies. Several traits of the model are presented below:

- The model uses data driven approaches as opposed to physics-based methods to make real-time glycan predictions in monoclonal antibodies.
- The model is independent of the AMBIC and CDOptiCHO cell lines. Compared to the other available models, change of model parameters is not necessary to predict different cell lines.
- It is agnostic to the change in bioreactor conditions like temperature, pressure, and pH.
- It is independent of the addition of supplements like Ammonium Chloride, Asn, Gln.
- The model is capable of predicting multiple outputs like the 9-glycan relative distribution, 3 grouped glycan relative distribution and percentage Fucose level of the glycan distribution.
- It is also capable of handling multiple inputs like a time series data of 7 different variables or just the time series data of the antibody concentration profile.
- The static and dynamic models can be easily retrained when more data is available.
- More Variables can easily be added to or dropped from both static and dynamic models as illustrated in the modified static model.
- The stability and performance of the model can be increased very easily just by retraining it on more dataset without altering the main framework.

This model can be used for the following purposes:

- It can be used to design an effective controller to manipulate glycosylation in Monoclonal Antibodies: Getting real time prediction of the final glycan distribution will help to design a controller to keep the glycan distribution consistent from batch to batch.
- It can save days of manufacturing time: Accurate online predictions of glycosylation will help to save a huge amount of time and resources by stopping the process prematurely if

the final product quality has gone beyond the desired range and it is not possible to be manipulated within range by any means.

Future Work

The following ideas can be implemented to increase the robustness and the generalization ability of the model:

- The model can be trained on other cell line data as well. This will increase the generalization ability of the model.
- The dynamic model currently predicts only the antibody concentration profile but it can easily be used to predict all the 7 variables with enough experimental data. However, with the current volume of data, the model lost its stability when it was made to predict all the 7 variables. Hence, the static model needed to be modified to deal with that restriction.
- The model can be made more robust by training it using more data where the glycan distribution has gone out of range. Currently most of the data points have the glycan distribution within the specified ranges.
- The model at present is more biased towards the CDOptiCHO cell line since it has been trained on twice the amount of CDOptiCHO data than AMBIC data.
- Other variables like pH, temperature, dO2, media supplements can be incorporated in the model very easily. Most of them were neglected because either they are held constant throughout the experiment or they have very less influence on the glycan distribution.

Comparing the dynamic approach to a physics based approach

In this section, a comparison between the data driven dynamic model described above and a physics based dynamic model has been discussed. The physics-based model has been derived from the dissertation submitted to the Faculty of the University of Delaware by Devesh Radhakrishnan [5]. The model is the macroscopic part of the original multiscale model. The lactate equation of the model has been modified to make it work in python's ode solver, solve ivp, which cannot impose a non-negativity constraint on its arguments like MATLAB's built in ode solver. The key variables in this model are as follows:

- μ: Cell growth rate
- μ_d: Cell death rate
- Glc: Glucose Concentration
- q_{glc}: Specific consumption rate for glucose
- m_{glc}: Maintenance coefficient for glucose
- Gln: Glutamine Concentration
- q_{gln}: Specific consumption rate for glutamine
- m_{gln}: Maintenance coefficient for glutamine
- Lac: Lactate Concentration
- q_{Lac}: Specific lactate production rate
- q_{cons}: Specific lactate consumption rate
- Amm: Ammonia Concentration
- $q_{Amm:}$ Specific ammonia production rate
- X_v: Viable cell density
- X_t: Total cell density
- K_{lysis}: Cell lysis rate
- Mab: Antibody concentration
- q_{Mab}: specific antibody production rate

The parameters used in the model are as follows:

- Yield coefficient of biomass on glucose, $Y_{X/Glc}$, [cells/ mM] 1.40 x 109
- Yield coefficient of biomass on glutamine, $Y_{X/Gln}$, [cells/ mM] 2.70 x 109
- Yield coefficient of biomass on lactate, $Y_{X/Lac}$, [cells/ mM] 6.53 x 107
- Yield coefficient of ammonia on glutamine, $Y_{Amm/Gln}$, [mM/ mM] 0.63
- Yield coefficient of lactate on glucose, $Y_{Lac/Glc}$, [mM/ mM] 1.30
- Yield coefficient of mAb on glucose, $Y_{MAb/Glc}$, [g/L/ mM] 5.55 x 10-3

- Constant for glutamine degradation, $K_{d,Gln}$, [hour -1] 9.60 x 10-3
- Monod constant for glucose, K_{Glc}, [mM] 0.14
- Monod constant for lactate, K_{Lac}, [mM] 0.25
- Monod constant for glutamine, K_{Gln}, [mM] 0.025
- Constant for lactate inhibition, $K_{I,Lac}$, [mM] 171.76
- Constant for ammonia inhibition, $K_{I,Amm}$, [mM] 28.48
- Cell lysis rate, K_{lysis}, [hour -1] 0.02 – 0.06
- Glutamine maintenance coefficient, mgln, [mM-hour -1 /cells] 4.25 x 10-15
- Constant for glucose maintenance coefficient, a0, [mM-hour -1 /cells] 2.25 x 10-10
- Constant for glucose maintenance coefficient, a1, [mM] 39.65
- Maximum growth rate (exponential), μ_{max1}, [hour -1] 0.03
- Maximum growth rate (stationary), μ_{max2}, [hour -1] 6.50 x 10-3
- Maximum death rate, $\mu_{d,max}$, [hour -1] 0.042
- Death rate constant, kd0, [hour -1] 4.54 x 10-4
- Death rate constant, kd1, [hour -1] 5.00 x 10-3
- Self-defined constant, k_s, [unitless]-0.08

The model equations are as follows:

$$\mu = \frac{\mu_{max,1} * Glc * Gln * K_{i,Lac} * K_{i,Amm}}{(K_{Glc} + Glc) * (K_{Gln} + Gln) * \left(K_{i,Lac} + Lac\right) * (K_{i,Amm} + Amm)}$$

$$\mu_d = \mu_{d,max} * \frac{k_{d,0}}{k_{d,1} + \mu}$$

$$\frac{dGlc}{dt} = -q_{glc} * Xv$$

$$q_{glc} = \frac{\mu}{Y_{x,glc}} + m_{glc}$$

$$m_{glc} = a0 * \frac{Glc}{a1 + Glc}$$

$$\frac{dGln}{dt} = -q_{Gln} * Xv - K_{d,Gln} * Gln$$

$$q_{gln} = \frac{\mu}{Y_{x,gln}} + m_{gln}$$

$$\frac{dLac}{dt} = q_{Lac} * Xv - q_{cons} * Xv * \frac{Lac}{k_s * Glc}$$

$$q_{Lac} = Y_{LacGlc} * q_{Glc}$$

$$\frac{dAmm}{dt} = q_{Amm} * Xv + K_{d,gln} * Gln$$

$$q_{Amm} = Y_{Amm,Gln} * q_{Gln}$$

$$\frac{dMab}{dt} = q_{Mab} * Xv$$

$$q_{Mab} = Y_{Mab,Glc} * q_{Glc}$$

These equations were solved using scipy's solveivp ode solver. The initial values for the variables were as follows:

- Glc0 = 28 mM [Initial Glucose Concentration]
- Gln0 = 4 mM [Initial Glutamine Concentration]
- Lac0 = 0 mM [Initial Lactate Concentration]
- Amm0 = 0 mM [Initial Ammonia Concentration]
- Mab0 = 0 g/Ltr [Initial Antibody Concentration]
- Xv0 = 0.5e9 cells/Ltr [Initial Viable Cell Density]
- Xt0 = 0.5e9 cells/Ltr [Initial Total Cell Density]

The concentration profile of Glucose, Glutamine, Lactate and Ammonia obtained after solving the equations is given in Figure 23. The total and viable cell density is given in Figure 24 and the antibody concentration profile is given in Figure 25. While the model helps to understand the underlying physics behind glycosylation, it lacks the following important traits:

- It fails to adapt to different cell lines. All the parameters of the model need to be changed if the cell line changes

- It fails to adapt to different bioreactor conditions. Variables like temperature, pH, and dissolved oxygen are not included in the model. A slight change in any of those results in change in concentration profile of the variables. Thus, the model parameters need to be readjusted again,
- It does not consider the previous time steps and makes the same predictions for all cases.

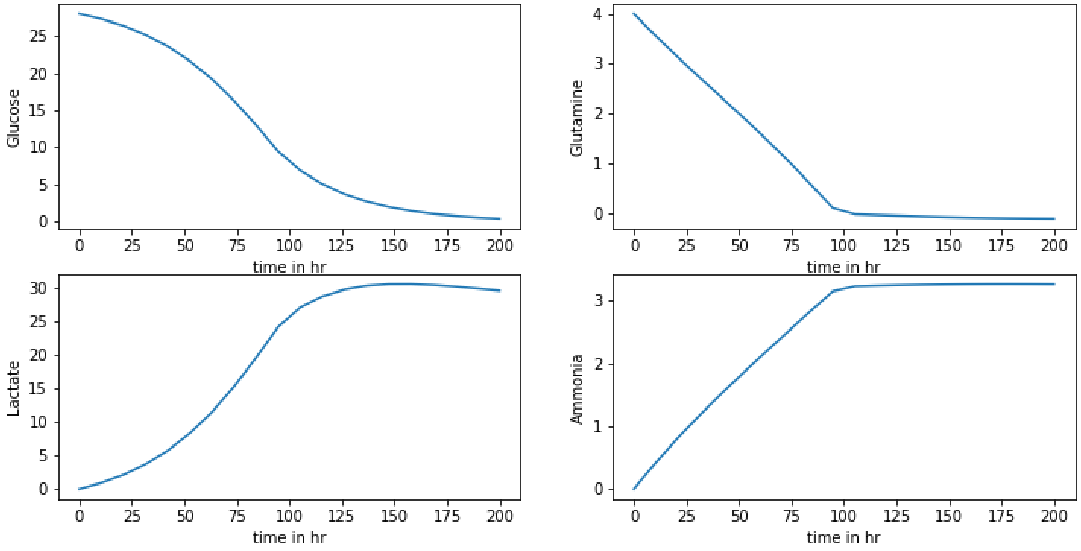


Figure 23: Media Concentration Profile given by the physics based dynamic model

On the other hand, the dynamic approach discussed above has the following advantages over this physics-based model:

- It is independent of the AMBIC and CDOptiCHO cell line. More cell lines can be added very easily to the model. It just needs to be trained on the cell line data. Thus, no change in model parameters is required to predict different cell lines.
- It considers previous time steps to make predictions.

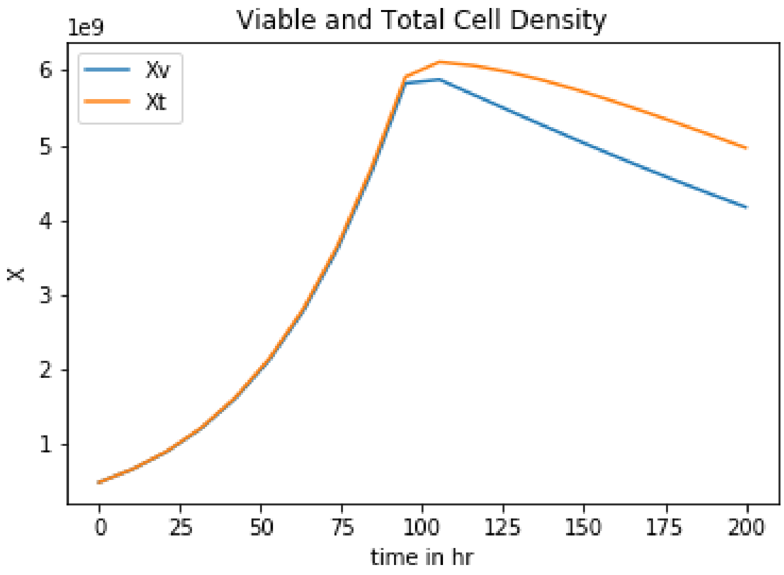


Figure 24: Total and Viable Cell Density

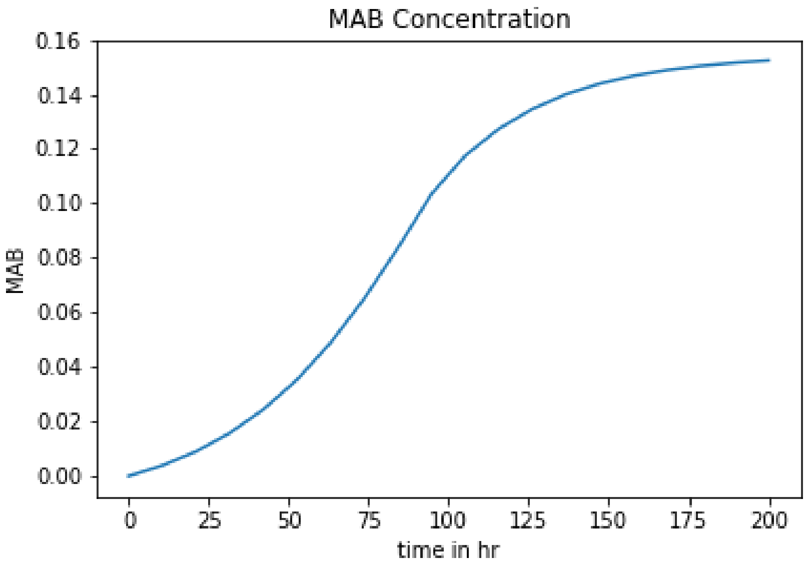


Figure 25: Antibody Concentration Profile

- It can handle slight changes in bioreactor conditions. Variables like temperature, dissolved oxygen, pH can be easily incorporated in the model.

However, the dynamic approach is completely data driven and it fails to explain the underlying physics behind glycosylation.

References

[1] A. S. a. U. S. Ralf Otto. [Online]. Available: https://www.mckinsey.com/industries/pharmaceuticals-and-medical-products/our-insights/rapid-growth-in-biopharma.

[2] [Online]. Available: https://biopharmadealmakers.nature.com/users/9880-biopharma-dealmakers/posts/53687-moving-up-with-the-monoclonals.

[3] D. Neri, "What Is a Biosimilar?," in *Frontiers of Gastrointestinal Research*, 2015.

[4] M. M. S. Amand, "TOWARD ONLINE QUALITY CONTROL DURING BIOPHARMACEUTICAL PRODUCTION," 2013.

[5] D. Radhakrishnan, "MODELING, ESTIMATION, AND CONTROL OF GLYCOSYLATION IN," 2016.

[6] "Encyclopædia Britannica, Inc," [Online]. Available: https://www.britannica.com/science/antibody.

[7] "Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Monoclonal_antibody.

[8] S. K. Yoon and S. L. Choi, "Effect of culture pH on erythropoietin production by Chinese hamster ovary cells grown in suspension at 32.5 and 37.0 degrees C".

[9] J. P. Kunkel and D. C. Jan, "Dissolved oxygen concentration in serum-free continuous culture affects N-linked glycosylation of a monoclonal antibody".

[10] "What is OPC?," [Online]. Available: https://opcdatahub.com/WhatIsOPC.html.

[11] D. Kominek, "OPC: The Ins and Outs to What It’s About".

[12] T. Yiu, "Understanding Neural Networks," [Online]. Available: https://towardsdatascience.com/understanding-neural-networks-19020b758230.

[13] "scikit-learn: machine learning in Python — scikit-learn 0.21.3 documentation," [Online]. Available: https://scikit-learn.org/stable/.

[14] S. Banerjee, "An Introduction to Recurrent Neural Networks," [Online]. Available: https://medium.com/explore-artificial-intelligence/an-introduction-to-recurrent-neural-networks-72c97bf0912.

[15] "Understanding LSTM Networks -- colah's blog," [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[16] "TensorFlow," [Online]. Available: https://www.tensorflow.org/.