TECHNISCHE UNIVERSITÄT WIEN
Vienna University of Technology

DIPLOMARBEIT

Automating pH Adjustment by Robotic Workflow and Supervised Machine Learning

Ausgeführt zum Zwecke der Erlangung des akademischen Grades eines

Diplom-Ingenieurs

im Rahmen des Studiums

Technische Chemie

eingereicht von

Philipp Müller-Bischof, BSc.

Matrikelnummer 01346695

Freiheitsstrasse 51

2331-Vösendorf

Ausgeführt am Institut für Chemische Technologien und Analytik
der Fakultät für Technische Chemie der Technischen Universität Wien
in Zusammenarbeit mit University of Cambridge, Department of Chemical Engineering

Betreuung (TU Wien):
Ao. Univ. Prof. Dipl.-Ing. Dr. techn. Egon Erwin Rosenberg

Betreuung (University of Cambridge):
Dipl. Ing. Alexander Pomberger
Prof. Alexei Lapkin

Vösendorf, Februar 2024

Acknowledgments

I want to thank Prof. Alexei Lapkin from the University of Cambridge for the chance to be part of his research group.

I also want to thank Prof. Erwin Rosenberg from the TU Wien, who agreed to accept a review of this thesis and provided feedback during the project.

I also want to thank Alexander Pomberger whom I met in the first days of my chemistry studies and who became a friend over the years. It was him who recruited me for the present work, and he was my direct contact person for this project. He took the time whenever it was needed and provided great support for this master thesis.

Thanks to my wife Martina and my children Henrik and Kristin, who have shown a lot of understanding in some turbulent times.

Abstract

The adjustment of pH value is an important task in many industrial processes, such as in life science research or formulated products. Several relevant buffer systems consist of mixtures of multiple-buffering substances. While single-buffer systems can be described via the Henderson-Hasselbalch equation, there is no known mathematical method to directly resolve the calculation of pH for multi-buffered polyprotic systems. The objective of this work is to provide an approach that is different to the common practice of pH adjustment like manual adjustment or adjustment by PID controllers.

Since the application of machine learning for chemical challenges is a topic of great current interest, we hypothesize that it might also be beneficial for predicting and adjusting the pH value of complex samples.

This master thesis provides a purely data driven machine-learning approach in form of an iterative closed-loop optimization process for pH adjustment of multi-buffered polyprotic systems. The software was written in Python. A personal computer with 16 GB RAM and a 3.4 GHz Processor was used for both, the programming and optimization work. Commonly used surrogate models, like artificial neural network, random forest, linear regression and Gaussian process were tested in order to compare the overall performance on solving the present task. The benchmarking was based on the efficiency of titration towards the pre-defined target pH value. Efficiency means a small number of iteration cycles in this case. All models were able to solve the problem, with the Gaussian process requiring the least number of loop runs.

Zusammenfassung

In vielen industriellen Prozessen ist es unerlässlich, pH-Werte zu adjustieren. Während großtechnische Lösungen, beispielsweise in der Abwasserneutralisierung, oft über PID-Regler realisiert werden, ist dieser Ansatz bei kleinen Batches im Milliliter-Bereich nicht zielführend. Häufig liegen multigepufferte, polyprotische Systeme vor, die mathematisch nicht über die Henderson-Hasselbalch-Gleichung beschrieben werden können.

In den letzten Jahren hat die Zahl der Publikationen, die sich direkt oder indirekt mit dem Thema “maschinelles Lernen” oder “künstliche Intelligenz” beschäftigen, stark zugenommen. Daher lag es nahe, diesen Ansatz auch für die effiziente Einstellung des pH-Wertes zu verwenden. In der vorliegenden Diplomarbeit wird ein rein datenbasierter Lösungsansatz präsentiert, um pH-Werte von multigepufferten, polyprotischen Puffersystemen mit größtmöglicher Effizienz hinsichtlich Mess- und Regelungsaufwand zu bewerkstelligen. Die Software wurde in der Programmiersprache Python geschrieben. Sowohl die Programmierung als auch die Optimierung der Modelle erfolgte auf einem PC mit 16 GB RAM und einem 3.4 GHz Prozessor. Als Modelle wurden künstliche neuronale Netzwerke, random forest, lineare Regression und der Gaussprozess herangezogen und auf Ihre Leistungsfähigkeit überprüft, die vorliegende Problemstellung zu lösen. Zur Feststellung der Effizienz eines Modells wurden die benötigten Iterationen bis zum Erreichen des Zielwertes herangezogen. Die erfolgreiche Adjustierung des pH-Wertes gelang mit allen erwähnten Modellen, wobei sich der Gaussprozess als effizientestes Modell herausstellte.

List of Abbreviations

ANN	artificial neural network
AF	activation function
ELU	exponential linear unit
GP	gaussian process
GPR	gaussian process regression
HPO	hyperparameter optimization
ML	machine learning
RA	regression analysis
RBF	radial basis function
ReLU	rectified linear unit
RF	random forest
SVM	support vector machines

Table of Contents

1 Introduction

A brief historical introduction followed by definitions of pH-value, -measurement, and -adjustment, as well as buffer-systems will lay the chemical groundwork for the automated adjustment procedure. Further introduction of automation of chemical experiments and the role of machine learning in particular will finish this chapter.

1.1 The pH value

In 1887 S. Arrhenius, a Swedish scientist, introduced that acids and bases are substances that dissociate in water to yield electrically charged atoms or molecules, called ions. Acids dissociate in water to yield hydrogen ions (H^+) (Equation 1) and bases ionize in water to yield hydroxide ions (OH^-) (Equation 2).

$$HCl + H_2O \xrightarrow{dissociation} H_3O^+ + Cl^- \qquad \textit{Equation 1}$$

$$NaOH \xrightarrow{dissociation} Na^+ + OH^- \qquad \textit{Equation 2}$$

Arrhenius realized that the acidic and basic properties are depending on the hydrogen ion concentration and hydroxide ion concentration, respectively. [1] In 1923 the chemists J. Brønsted and T. Lowry added to Arrhenius' theory that a compound that can transfer a proton to another compound is an acid, and the compound that accepts the proton is a base. [2] Finally, G. Lewis provided the most general definition for acids and bases in 1923. According to his theory an acid is regarded as a compound which, in a chemical reaction, can attach itself to an unshared pair of electrons in another molecule (Lewis-acid). The molecule with an available electron pair is called a base. [3] The Danish biochemist S. P. L. Sørensen proposed pH in 1909 first as the negative decadic logarithm of the oxonium ion concentration ($[H_3O^+]$) or hydrogen ion concentration ($[H^+]$) (Equation 3) and later in terms of activity (Equation 4). The activity incorporates non-ideal interactions (solvent-solvent, solvent-solute, and solute-solute), which are important parameters in more concentrated systems.

$$pH = -\log_{10}\left[\frac{c_{H^+}}{mol\ L^{-1}}\right] \equiv -\log_{10}\left(\frac{c_{H^+}}{c^0}\right) \qquad \textit{Equation 3}$$

Where c_{H^+} is the concentration of hydrogen ions, and c^0 is the concentration in the standard-state.

$$pH = -\log_{10}(a_{m,H^+}) = -\log_{10}\left(\frac{m_{H^+}\gamma_{m,H^+}}{m^0}\right) \qquad \textit{Equation 4}$$

Where a_{m,H^+} is the temperature-independent molality (dimensionless), m_{H^+} is the molality (mol kg^{-1}) of the hydrogen ions, m^0 is the unit molality, and γ_{m,H^+} is the activity coefficient on the molality basis (dimensionless) [4], [5]. H^+ indicates the bare proton but should be interpreted as the sum of all hydrated proton species. Some of them are large clusters like the "Zundel cation" $H_5O_2^+$ and the "Eigen cation" $(H_3O^+)(H_2O)_3$. [6]

Commonly used pH scale is shown in Figure 1. Values below pH 7 are increasingly acidic and values above 7 are increasingly basic. A pH value of 7 is considered neutral. [7]

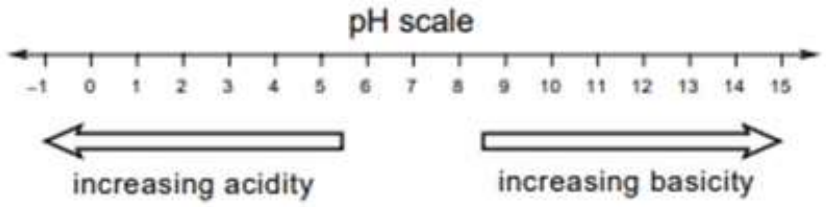

Figure 1: Commonly used pH scale with increasing acidity left from pH 7 and increasing basicity right from pH 7.

1.2 pH measurement

Commonly used state-of-the-art pH meters consisting of a combination of a glass membrane and a reference electrode ($Hg|Hg_2Cl_2$ or Ag|AgCl electrode, see Figure 2 [8]). Equation 5 leads to the pH of an aqueous solution of interest (pH(X)).

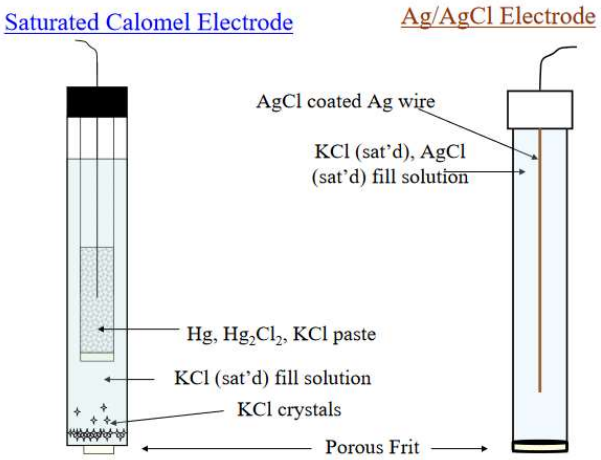

Figure 2: Schematic of a calomel electrode on the left side and a Ag/AgCl electrode on the right side [8]

$$pH(X) = \frac{E_X - E_{SS}}{\frac{RT}{F}\ln(10)} + pH_{SS} \qquad \textit{Equation 5}$$

Where E_X is the experimental cell potential measured for X (compound of interest), E_{SS} is the experimental cell potential of a secondary standard (or E_{PS} primary standard), R is the universal gas constant, T the temperature (in Kelvin), and F the Faraday constant.[6]

Figure 3 shows different state-of-the-art pH-electrodes for laboratory and industrial use. [9]

Figure 3: Examples of pH-electrodes for laboratory and industrial use [9]

1.3 pH buffer

Systems emerged during evolution that can resist pH changes in biological systems – so called "pH buffer" or simply "buffer". Sørensen was the first who mentioned the word buffer in 1909. [10] In buffered systems, the addition of strong acids or bases leads to smaller pH changes compared to unbuffered systems [11] (Figures 4, 5). As a result, the pH of the unbuffered solution can change considerably in contrast to the buffered solution, whose pH stays relatively stable (Figure 5).

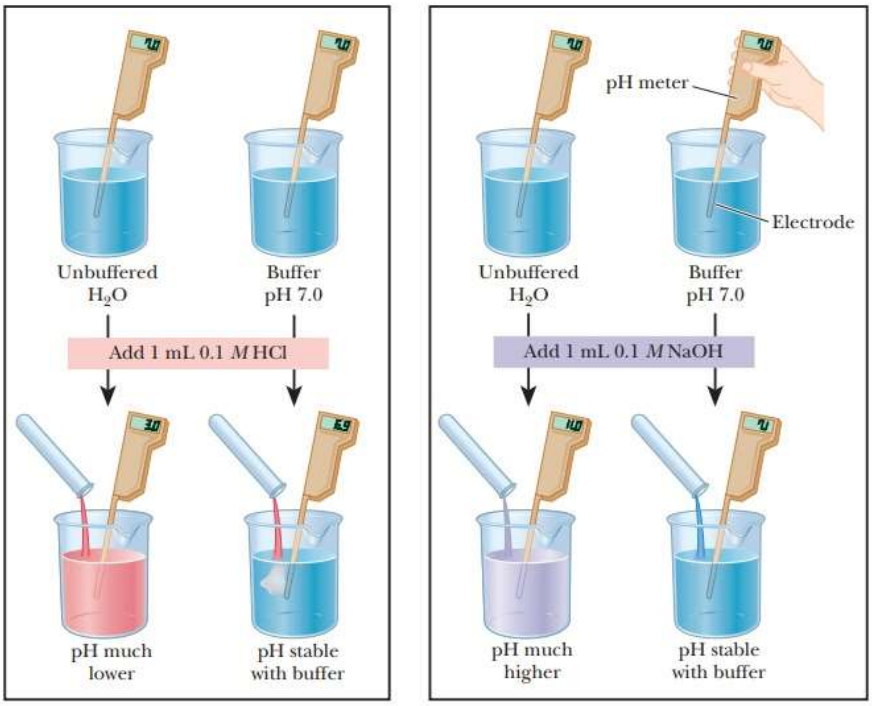

Figure 4: Reaction of buffered vs. unbuffered solution upon acid/base adding. The buffered solutions have a far less change of pH after adding acid or base as opposed to the unbuffered solution. [12]

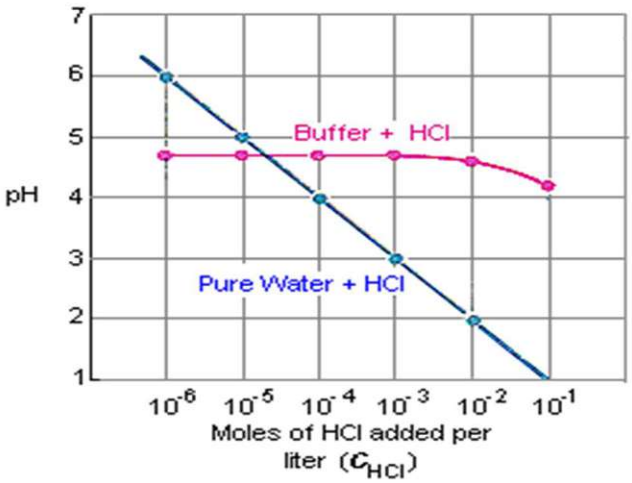

Figure 5: Graphic representation of the pH of a buffered vs. an unbuffered system after adding HCl. The pH value of the unbuffered solution constantly decreases while the buffered solution is relatively stable before the buffer capacity is exceeded. [13]

In general, a buffer solution is an aqueous solution that consists of a weak acid and its conjugate base or a weak base and its conjugate acid. [14] After adding a small amount

of a strong acid, the free hydrogen ions (H^+) are reacting with the conjugate base from the buffer. That leads to some resistance of pH change. [15]

Buffers have an important meaning in biological systems as well as in chemistry in general and in formulated products in particular. A common example for an in vivo buffer system is human blood which contains a bicarbonate buffer system. [14]

A commonly used buffer in chemistry is the acetic acid-acetate buffer. The equations below explain its function. Equation 6 describes the absorbance effect of OH^- ions and Equation 7 shows the absorbance effect of H^+ ions. [15]

$$CH_3COOH + OH^- \rightleftarrows CH_3COO^- + H_2O \qquad \textit{Equation 6}$$

$$CH_3COO^- + H^+ \rightleftarrows CH_3COOH \qquad \textit{Equation 7}$$

Even if a strong acid or base is added, the pH value only changes slightly. This observation is called buffering effect. A 1:1 equimolar ratio of CH_3COO^-/CH_3COOH results in a pH value corresponding to the pKa of the acetic acid, which is 4.75 at 25 °C. In the range of $0.1 \leq \frac{c_{A^-}}{c_{HA}} \leq 10$ the buffering effect can be observed. Applying the decadic logarithm to these ratios leads to a pH area of ±1 away from pKa.

The buffer capacity (β) is defined as the amount of acid or base (in mol) that must be added to change the pH by ±1. [5] The formal expression is shown in Equation 8.

$$\beta = \frac{n}{\Delta pH} \qquad \textit{Equation 8}$$

The calculation of buffer solutions can be derived from the law of mass action of the acid protolysis reaction (Equation 9, 10): [15]

$$HA + H_2O \rightleftarrows H_3O^+ + A^- \qquad \textit{Equation 9}$$

$$K_a = \frac{[H^+][A^-]}{[HA]} \qquad \textit{Equation 10}$$

Applying the logarithm to both sides leads to equation 11:

$$\log\ (K_a) = \log\ ([H^+]) + \log\left(\frac{[A^-]}{[HA]}\right) \qquad \textit{Equation 11}$$

Finally, inserting the mathematical definition of pH leads to Equation 12:

$$pH = pK_a + \log\left(\frac{[A^-]}{[HA]}\right) \qquad \textit{Equation 12}$$

Equation 11 is commonly known as “Henderson-Hasselbalch Equation”, which can be used to estimate the pH of a buffer solution. [11] But the Henderson-Hasselbalch equation has its limitations. It cannot be used to describe multiple buffered systems.

M. Nguyen, L. Kao and I. Kurtz provided a predictive mathematical formula to calculate the pH in a multiple-buffered aqueous solution. The method is based on the partitioning of the protons among various buffer pairs. The equilibrium [H^+] is solely calculated by the partitioning of the protons. [16] The work takes multiple-buffered systems into account but is limited to monoprotic chemicals.

1.4 pH adjustment

PID controllers are commonly used and accepted in industry. PID stands for proportional – integral – derivative, which are representing the three terms in the system. The PID control loop constantly calculates the discrepancy between the measured and the target value and controls the actuators accordingly. [17]

They are for example used for neutralizing wastewater in industrial continuous processes (Figure 6) [18], or for temperature control in industrial plants, only to name two.

The use of PID controllers can also be considered but was not within the scope of this work.

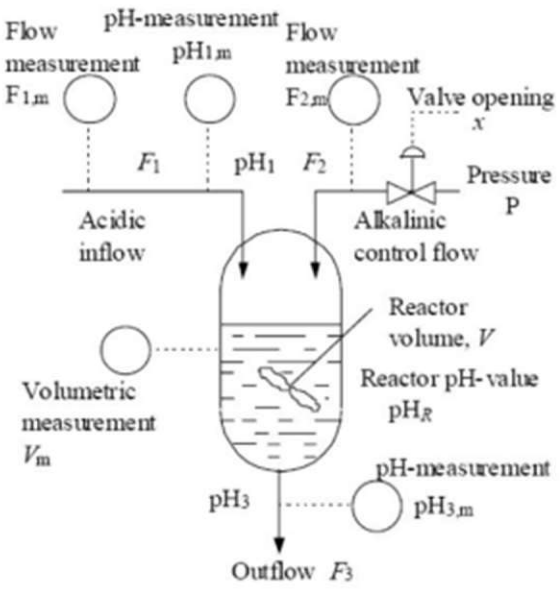

Figure 6: Schematic of a pH Neutralization Process. Strong base is used for neutralizing acidic liquid. The pH measurement pH_1 measures the inlet pH and measurement pH_3 the outlet. The control algorithm calculates the amount of base to be added and controls the valve accordingly. [18]

Machine learning has already been used in pH neutralization processes. M. Elarafi and S. Hisham showed that an artificial neural network outperformed a traditional PID controller. [19]

1.5 Automation of chemical experiments

The field of automation in chemistry dates back in the late 1960s derived from the demands in life sciences for more productive testing and analysis. Since then, the field expanded to chemical reactions, drug discovery, and material discovery for clean energy. Advances in both software and hardware made these systems much more robust and versatile. As examples of automated robotic experiments Figure 7 shows a fleet of systems with distinct sets of functions was developed by Bristol-Myers Squibb. [20]

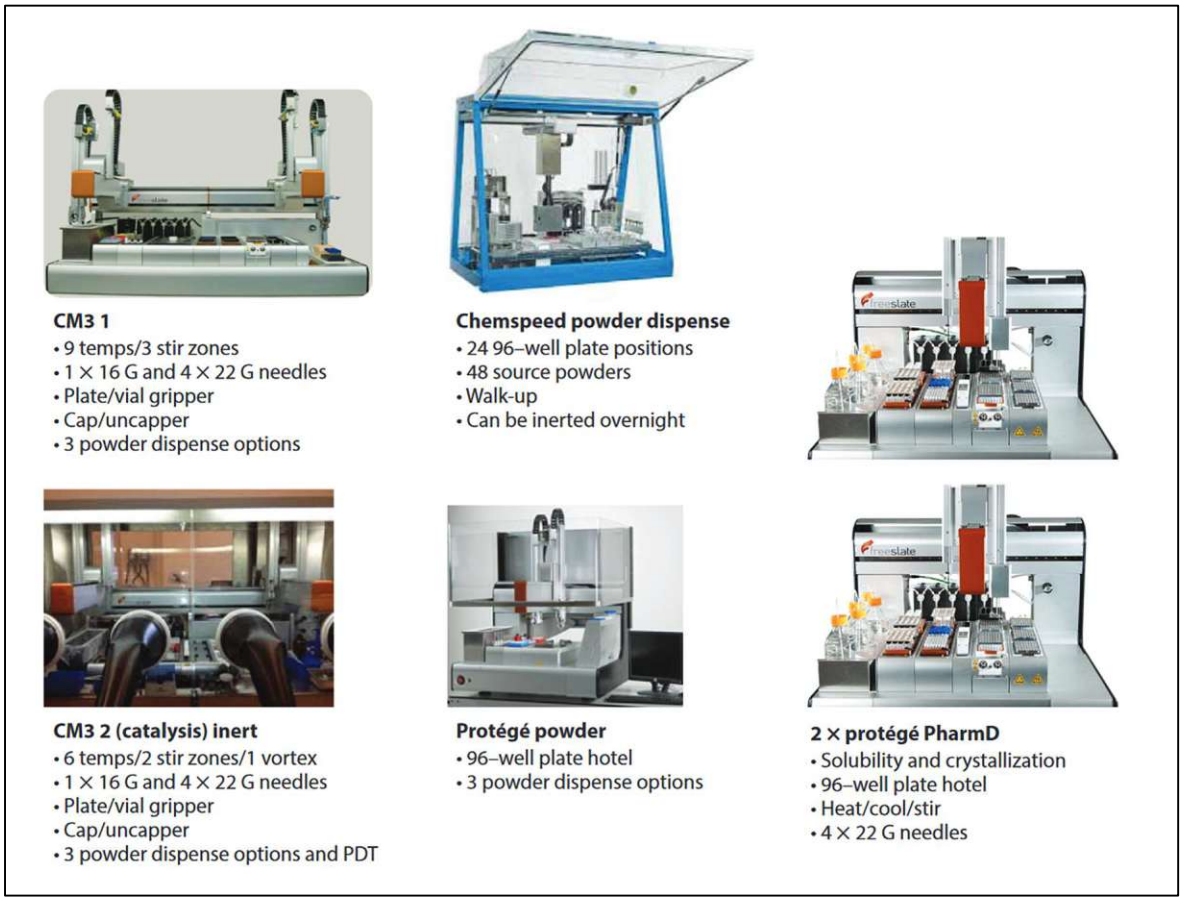

Figure 7: Examples of automated equipment used for chemical process development. Different abilities allow automated workflows for optimization or development processes. [20]

A. Aspuru-Guzik and K. Persson [21] came up with the idea of a platform-based approach to accelerate the material discovery process in 2018, and later extended by machine learning algorithms by M. Flores-Leonar *et al.* in 2020. [22]

The big difference to earlier automation tasks is that machine learning algorithms can make decisions, which enables them to manage tasks like autonomously synthesize, process and characterize organic components for example. [23]

As far as these MAPs (Materials Acceleration Platforms) already handle some of the mentioned tasks they are not fully autonomously up till now. It is an ongoing challenge to design fully autonomous robots to accelerate the research capabilities for new materials. [22]

1.6 Machine learning

A. Samuel is considered the inventor of machine learning (ML). It is defined as the field of study that gives computers the ability to learn without being explicitly programmed. It is said that the machine has learnt from its experience if its measurable performance in these tasks improves as it gains more and more experience in executing these tasks. The purpose of ML is to learn from relevant data and build models that describe the data structure.

ML has already moved into a wide variety of fields, like robotics [24], pattern recognition [25], computer games [26], traffic prediction [27], language processing [28], medical diagnosis (cancer detection) [29], E-mail spam filtering [30] and product recommendation [31] only to name a few. [32]

One commonly used methodology in ML is called “supervised learning” or “supervised machine learning”. It uses datasets as input-data to train algorithms and make predictions over an unseen area, once it has been fitted. [33]

Depending on the problem to solve, different mathematical models are better suited than others. Every ML prediction task is a two-step process. A learning step on the one hand and a prediction step on the other hand. Therefore, different models can be used. There are various commonly used algorithms: Decision tree, Gaussian process, support vector machine, or artificial neural network, only to name a few. [32], [34]

Today a so called “closed-loop optimization”, based on supervised ML, is widely used for chemical reaction optimization and has been first published by L. Coa, D. Russo and A. Lapkin [35] and C. Coley, N. Eyke and K. Jensen [36], [37]

Closed loop in this case means that the predicted pH value of the prior cycle acts as an additional new input for the following cycle if the earlier prediction does not meet the target pH value criteria.

1.8 Algorithms

1.8.1 Linear regression

Regression analysis (RA) is a way in mathematical statistics to generate an equation that can make predictions about the input data. Today RA is used in many scientific fields, like medicine, biology, agriculture, economics, engineering, sociology, geology, etc. [38]

For example, L. Müller-Wirtz *et al.* suggested a model to show the correlation between exhaled propofol concentration in plasma versus concentration in brain tissue. [39]

C. Doucouliagos and P. Laroche explored the economic impact of unions on productivity, using regression methods. [40]

Historically, linear regression was the first type of regression analysis that was widely and extensively used. It goes back to 1805, when Legendre published the earliest form – the least square method. [41]

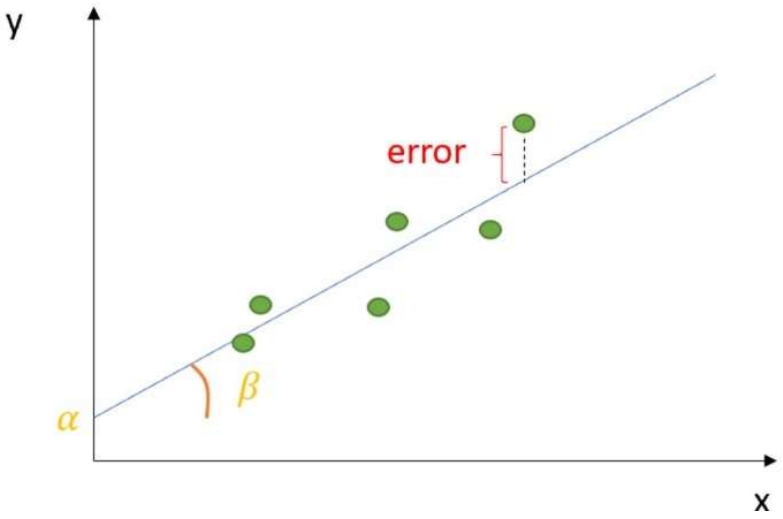

Figure 8: Graphical visualization of errors between datapoints (green dots) and regression function (blue line) for a linear regression model. [42]

The target is to minimize the sum of the squared residuals. Residual means the difference between a data point and the regression line (or prediction). A graphic representation can be found in Figure 8. A. Legendre used the method for his astronomical observations. [41]

Linear regression will be used to make a prediction or forecast of unseen datapoints by training the model on an observed set of data. Beside simple linear regression, where a linear relationship between two variables is observed (Equation 13), multiple linear regression (Equation 14) and nonlinear regression (Equation 15) also need to be mentioned. Multiple linear regression methods answer questions where one dependent variable (y) and more than one independent variables (x_1, x_2,⋯, x_n) are existing. The last type, nonlinear regression, assumes, that the relation between dependent variable (y) and independent variable (x) is not linear – like in exponential growth models for example. [38]

$$y = \beta_0 + \beta_1 x + \varepsilon \qquad \textit{Equation 13}$$

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon \qquad \textit{Equation 14}$$

$$y = \frac{\alpha}{1+e^{\beta t}} + \varepsilon \qquad \textit{Equation 15}$$

Where y is the dependent variable, x the independent variable, β_i are the regression coefficients and ε represents the error term. The regression line is calculated in regards of minimizing the error term ε, which represents the "closest" line to all data points. [38]

Interpretation of large datasets is often an issue. Principal component analysis (PCA) is the oldest and most widely used technique to reduce a higher dimensional dataset into a lower one, without losing relevant statistical information or patterns. Principal components are linear combinations of uncorrelated variables. [43], [44]

1.8.2 Random forest

Random forest can be used for classification and regression tasks. As the name already suggests the decision tree algorithm represents rules how data is split up and organized in a tree like structure. It belongs to the family of supervised learning algorithms. Each leaf represents different attributes, and each branch represents a value that the leaf can take. [32] Figure 9 shows an example for a classification task:

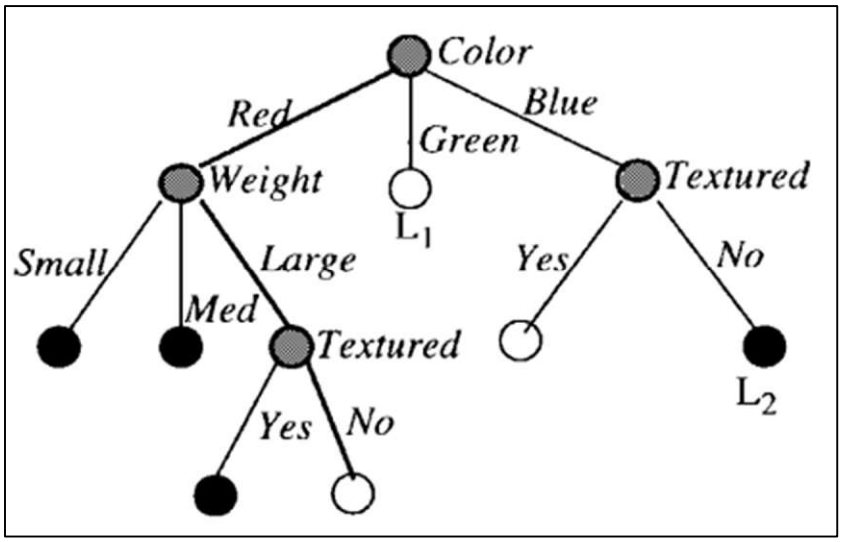

Figure 9: Decision tree example. Based on a tree, the leaves representing attributes and the branches representing the values for the respective attribute. [45]

If the value for the attribute "Color" is red, then the attribute "Weight" is queried next. On the other hand, if the color is blue, the attribute "Texture" is classified instead.

In case of numerical tasks, like pH adjustments, number ranges are used as attributes (Figure 10). The classification process is exactly the same as in the previous example.

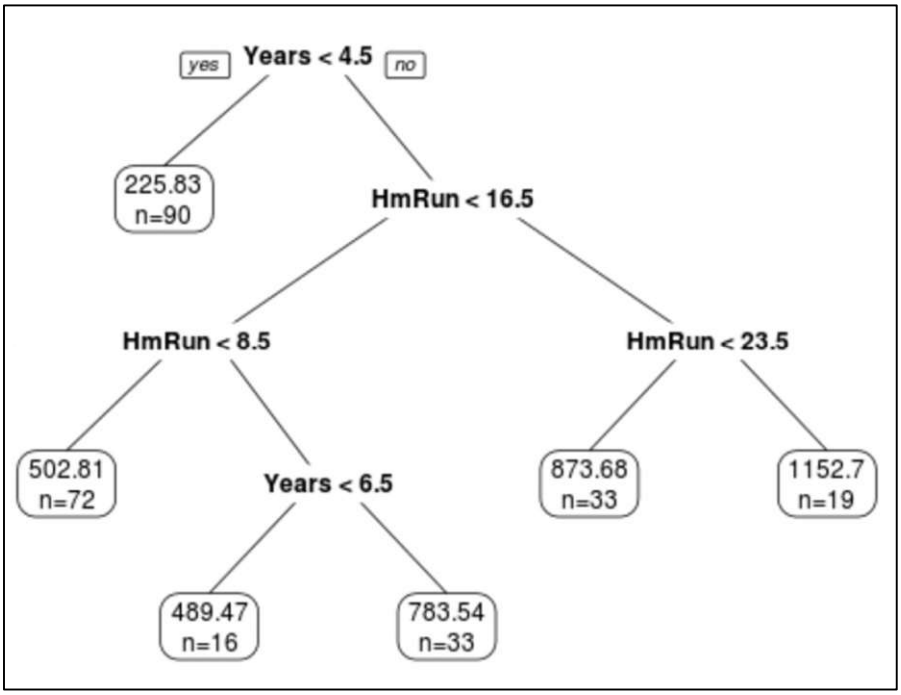

Figure 10: Regression task example to predict the salary of professional baseball players in terms of years of experience and number of home runs. [46]

The random forest model is a specific form of multiple decision trees (hence forest) that grew during the training phase. Each tree, also called estimator, uses different attribute hierarchy and values, which leads to different results.

The final prediction of the random forest is calculated as an average over all estimators, which is illustrated in Figure 11.

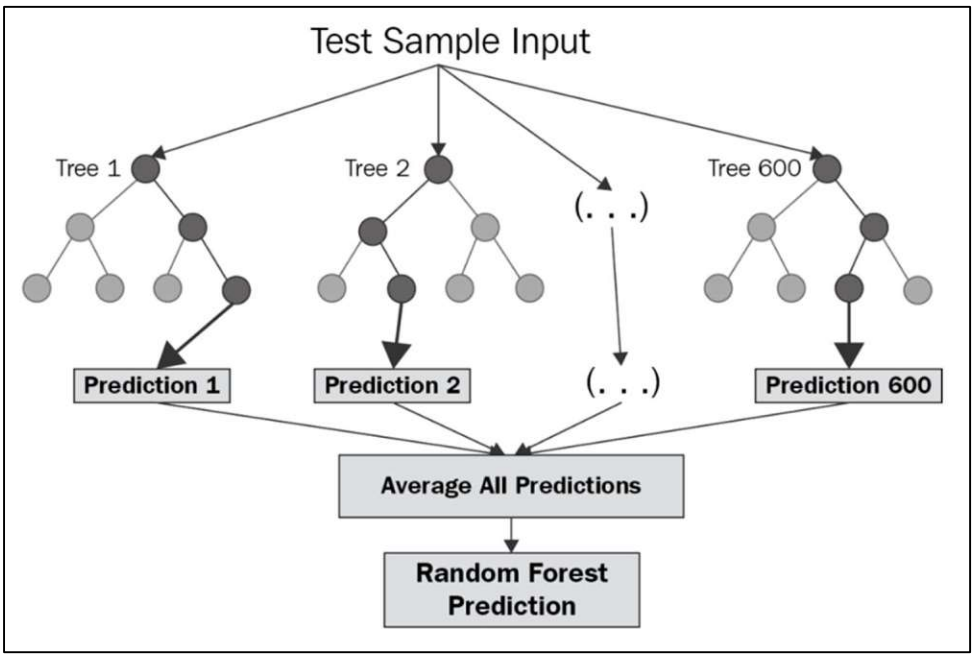

Figure 11: Random forest prediction procedure overview. The trees, or estimators, arise during the training phase of the algorithm. Different attribute hierarchy and values leads to different classification order and therefore different results. The final prediction is the result of an arithmetic mean calculation. [47]

1.8.3 Gaussian process regression

Gaussian process regression (GPR) is a non-parametric, Bayesian regression method. [48]

Non-parametric means that there is no predetermined form of the predictor. Bayesian means that prior knowledge (or a-priori knowledge) will be integrated in future predictions. [49]

GPR is placed into the class of linear smoothers, because the function f(x) is a linear combination of observed target values y. It predicts a posteriori Gaussian distribution for targets by observation of the input training data. In a nutshell, GPR predicts a Gaussian distribution at each datapoint (Figure 12). [50]

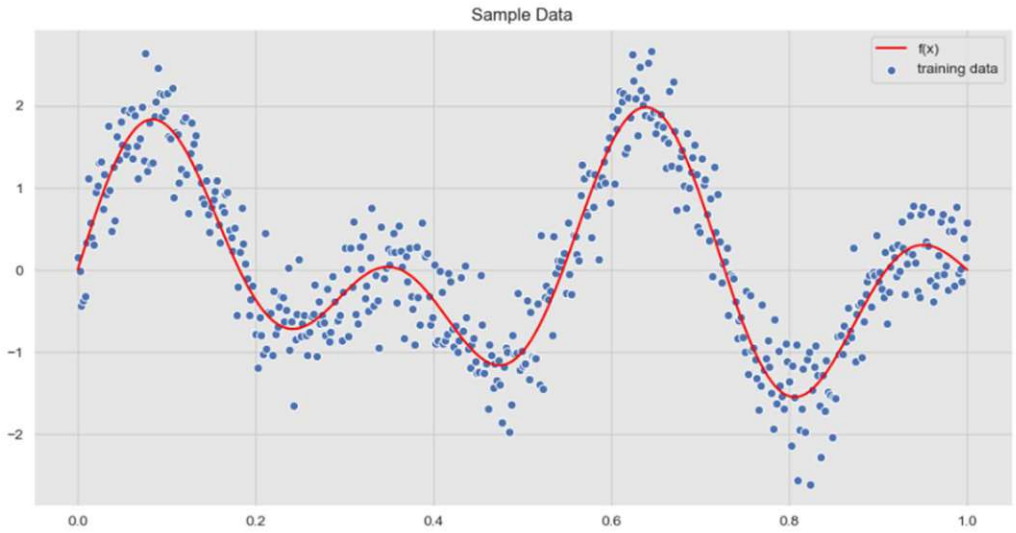

Figure 12: Visualization of a gaussian process fitted function on available training data [51]

Kernels are used to compute the covariance between datapoints. Covariance is a measure of the joint variability of two random variables. [52]

One big advantage of GPR is that kernels can be composed of simpler kernel functions to fit an assumed function, where e.g., prior knowledge of a relationship can be implemented. Figure 13 shows an example of different composed kernels for predicting atmospheric carbon dioxide (CO_2) over the years. [48]

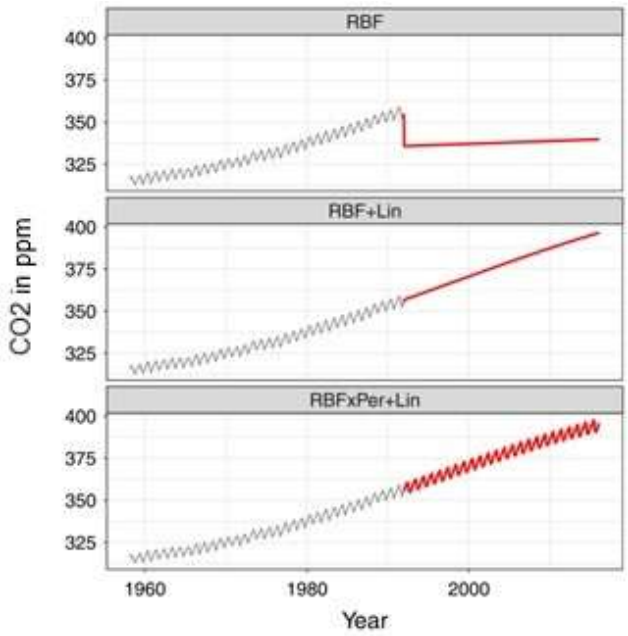

Figure 13: Kernel composition example for predicting atmospheric CO_2 over the years. [48] RBF stands for radial basis function, Lin stands for linear kernel, and Per stands for periodic kernel.

The single kernel functions can be simply added or multiplied to give a new complex kernel function containing characteristics of multiple kernels. Equation 16 shows an example for such a kernel used during this work.

$$K = C(1.0)*(RBF() + DotProduct()) \qquad \textit{Equation 16}$$

Parameter C is related to the kernel function of support vector machines (SVM) and is a measure of misclassification of training data. Lower values for C result in a smoother curve, whereas higher values of C aims for an exact classification of the training data. [53]

RBF stands for radial basis function that computes how close two datapoints are to each other. Equation 17 shows the mathematical calculation for two datapoints X_1 and X_2:

$$K(X_1, X_2) = \exp\left(-\frac{||X_1 - X_2||^2}{2\sigma^2}\right) \qquad \textit{Equation 17}$$

Where σ is the variance and the expression $||X_1 - X_2||$ the distance between X_1 and X_2. [54]

1.8.4 Artificial neural networks

Artificial Neural Networks (ANNs) mimic the process how the human brain operates. That is why the single decision nodes are called (artificial) neurons. Each of them can transmit a signal to another neuron, like synapses in biological systems. Each neuron typically has a weight, that can increase or decrease during the learning process – like the process we know from strengthening neural pathways in biological brains.

ANNs consists of at least three layers: one input layer, one or more hidden layers and one output layer. Each layer can contain any number of neurons. (Figure 14)

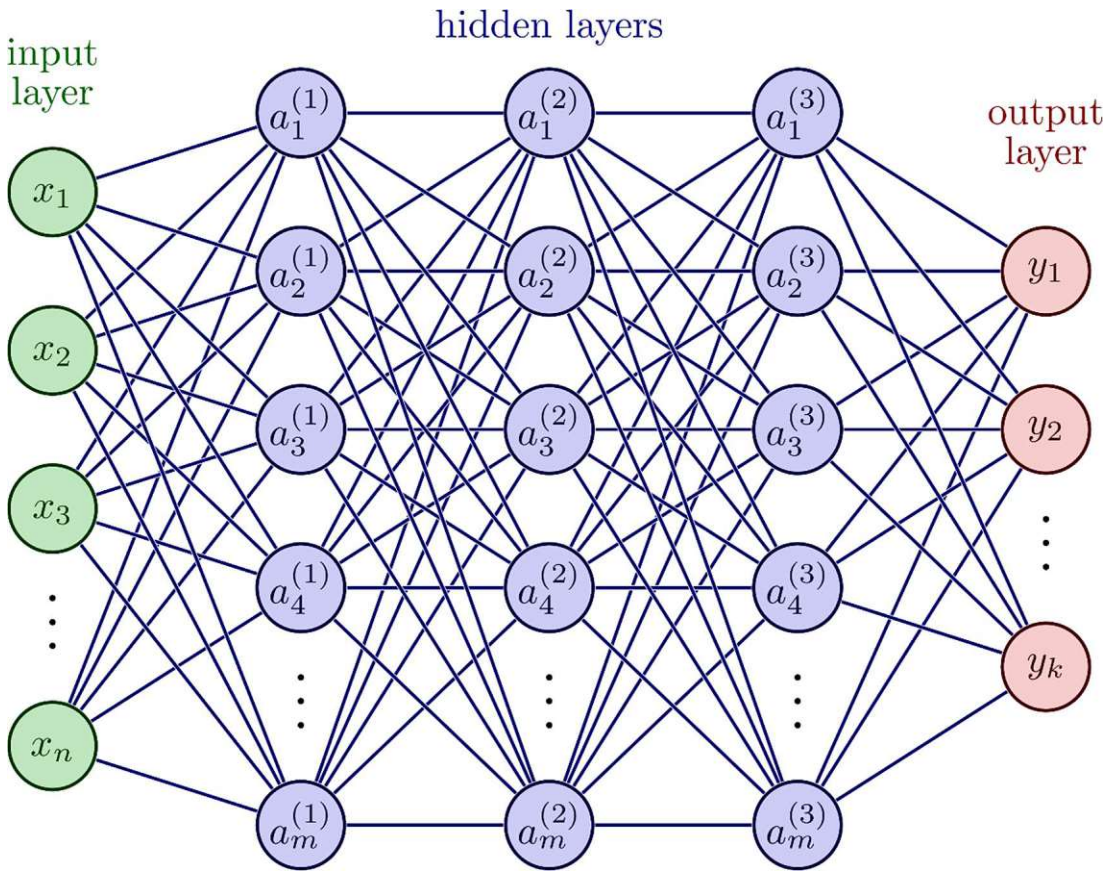

Figure 14: Schematic of an artificial neural network (ANN). The circles are called neurons. The "x" labelled circles are forming the input layer. The lines are representing the information flow from left to right through the hidden layer with the circles labelled with an "a". The "y" labelled circles form the output layer. [55]

The calculation of the output h_i is described by Equation 18 below:

$$h_i = \sigma\left(\sum_{j=1}^{N} V_{ij} x_j + T_i^{hid}\right) \qquad \textit{Equation 18}$$

Where σ is the activation function, N the number of input neurons, i the neuron in the hidden layer, V_{ij} the weights, x_j inputs to the input neuron, and T_i^{hid} the threshold terms of the hidden neurons. The purpose of the activation function σ is to generate nonlinearity on the one hand, and to prevent the ANN from paralysis by divergent neurons on the other hand. Different activation functions are available – among the most common ones are: sigmoid (or logistic), hyperbolic tangent, rectified linear unit and exponential linear unit (see appendix A for further details). [32], [56]

1.9 Hyperparameters

Hyperparameters are input parameters for machine learning algorithms, that directly affect the learning process and therefore the performance of the machine learning model. Optimal hyperparameter finding is crucial for ML algorithms, because they have a massive influence on how the algorithms behave during operation. These hyperparameters are variables that need to be tuned to design a good performing algorithm. This so called hyperparameter optimization (HPO) - or tuning - will be performed during the training phase. In practice different models will be trained with different hyperparameters and finally the prediction performance is compared with each other to find the optimum. There are two common methods of HPO: manual and automatic search. Manual search requires fundamental understanding for the underlying task, which is pH adjustment is this case. One needs to decide which parameters have more effect on the outcome than others and give more weight to them. Automatic search can be realized with different Blackbox-solutions. Two of them are grid search, which tries each combination of hyperparameters to find a global optimum, and random search which uses a random combination of hyperparameters in a set range,

as an example. Random search has efficiency advantages over the expensive grid search, but is not suitable for more complex tasks. [57]

Expensive in this case means, that grid search takes huge amounts of computational time. Therefore, manual optimization was chosen over an automatic method for the present work.

2 Methods

The following section describes the methods and experimental steps in order to decide which algorithm and hyperparameter setup will be used for further live testing.

2.1 Datasets

Different monoprotic and polyprotic substances were used as buffer examples. For each binary buffer system and mixtures, a separate data frame was generated. This led to 18 different datasets for investigating the efficiency and accuracy of different machine learning algorithms. The available datasets are listed in Table 1.

Table 1: Overview of the used binary buffer solutions, as well as the ratios and amount of titration datapoints.

ID	buffer system	ratio	total datapoints
1	acetate - citrate	1:1	97
2	acetate - citrate	1:2	127
3	acetate - citrate	2:1	87
4	acetate - KH_2PO_4	1:1	62
5	acetate - KH_2PO_4	1:2	68
6	acetate - KH_2PO_4	2:1	62
7	ammonium - acetate	1:1	62
8	ammonium - acetate	1:2	62
9	ammonium - acetate	2:1	62
10	ammonium - citrate	1:1	117
11	ammonium - citrate	1:2	117
12	ammonium - citrate	2:1	142
13	ammonium - KH_2PO_4	1:1	52
14	ammonium - KH_2PO_4	1:2	52
15	ammonium - KH_2PO_4	2:1	52
16	citrate - KH_2PO_4	1:1	127
17	citrate - KH_2PO_4	1:2	150
18	citrate - KH_2PO_4	2:1	137

The dataset consists of pH and pKa values for the buffers, the number of protons of the buffers, as well as concentration and volume of the acid and base for titration. Figure 15 shows the titration curve of the binary buffer system ammonium-acetate with a ratio of 2:1 as an example.

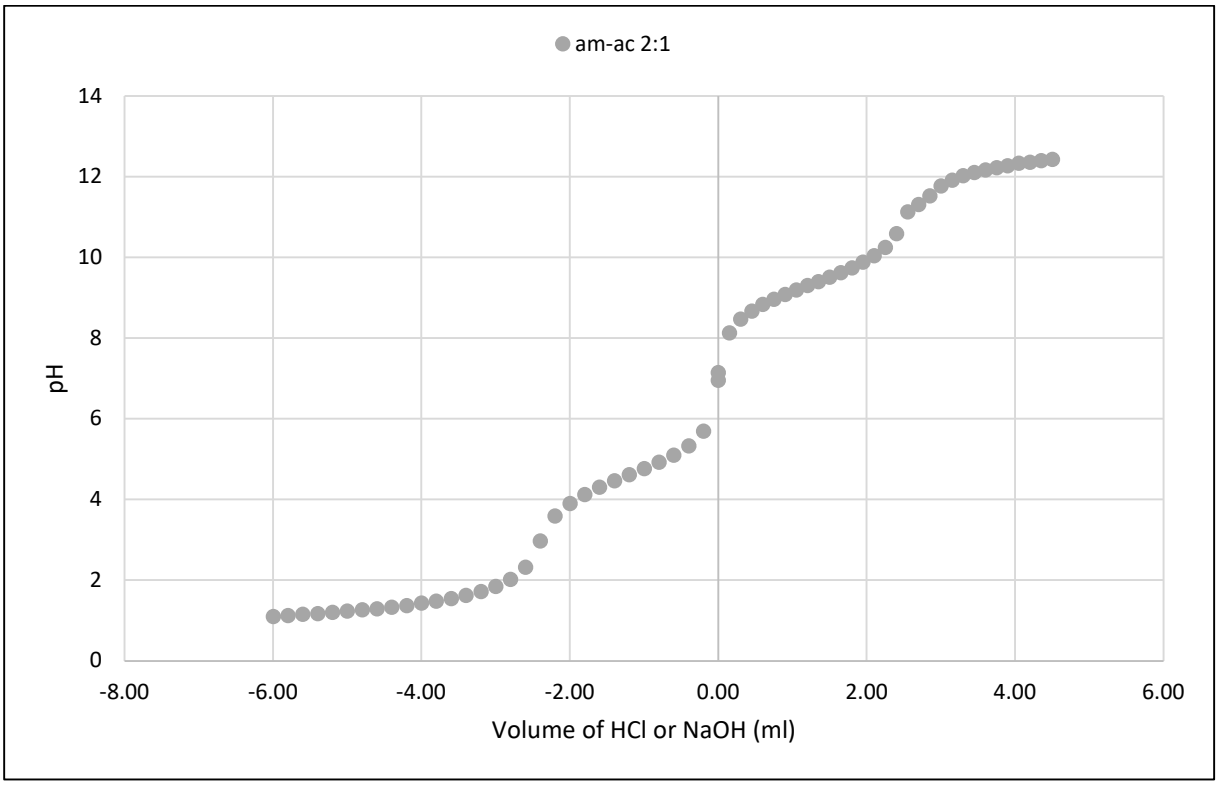

Figure 15: Titration curve for the binary buffer system ammonium-acetate with a ratio of 2:1. The addition of acid or base is indicated in ml on the x-axis. The addition of acid is indicated by a negative sign and the addition of base by a positive sign.

Only two of the total amounts of datapoints per dataset were used for the benchmark as an initial input (training-data) for the prediction of the pH value in the unexplored area. The others served as so called "test-data" to verify the prediction of the algorithm.

2.2 Benchmarking of different ML models within closed loop optimization

The system benchmark with optimized hyperparameters was done with regards to guided closed-loop optimization which is an iterative process (Figure 16). The binary buffer systems from Table 1 were used to perform a comparison of the different optimized algorithms to compare the results and decide which one is the best performing.

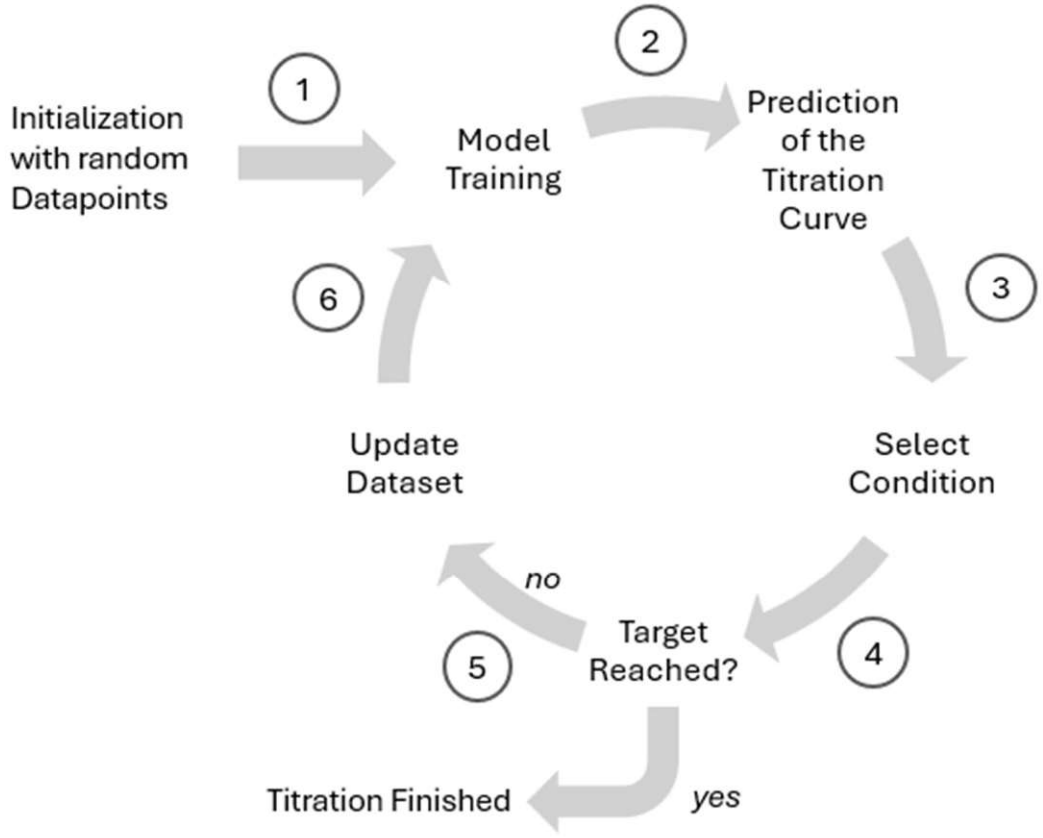

Figure 16: Principle of Closed-Loop Optimization. Initial input of datapoints (1), followed by training the model and predicting the titration curve based on the available information (2). Selecting the amount of acid or base depending on the position of measured pH on the curve (3). Verifying if the measured pH is within a range of ±0.2 of the target pH (4). If not, the dataset will be updated with the newly obtained datapoint (5) and finally the model will be retrained (6). This cycle continues until the target is met. [58]

During step 1, random datapoints serve as initial inputs and the ML model is trained on them. After that, the model is performing curve fitting, based on the available data in step 2 which leads to the prediction of the titration curve. Depending on the position of the measured pH value the amount of acid or base is predicted and added in step 3. During step 4 the algorithm compares the actual pH with the target pH. A deviation of ± 0.2 is acceptable. If the value is within this range the titration is finished, otherwise the new data is added to the existing training data (step 5) and the model will be retrained in step 6.

The number of iteration cycles to reach the target pH will be compared for different models, data representations and initialization strategies. The lower the average of iteration cycles, the better the prediction performance of the algorithm.

Figure 17 shows an example of the prediction of the four used models: artificial neural network (ANN), random forest (RF), linear regression and gaussian process (GP).

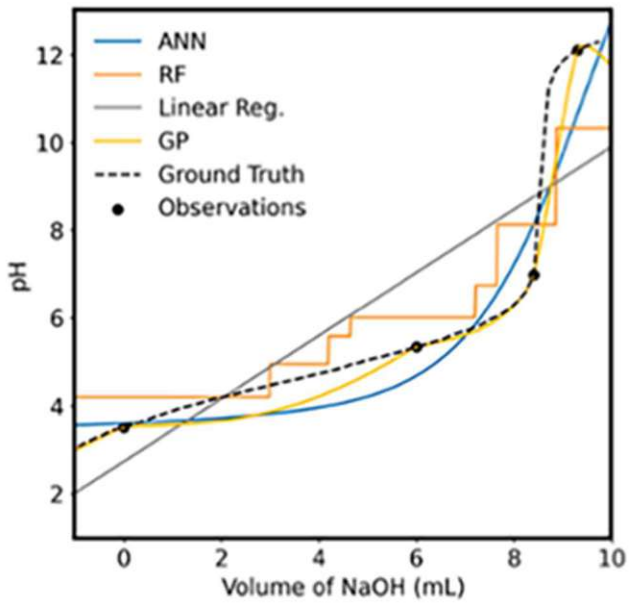

Figure 17: Comparison of prediction between artificial neural network (ANN), random forest (RF), linear regression, and gaussian process (GP) of an acetate-citrate 1:1 buffer system after four observations. [59]

3 Results and Discussion

The following chapter starts with the optimization results of the different models *(3.1 Optimization of the used models)*, followed by the benchmark of the optimized algorithms *(3.2 Benchmark results)* which leads to the final decision of the best performing algorithm.

Finally, there is a comparison between two different feature-sets as an input *(3.3 Feature-set comparison)*, to decide how much information the algorithm needs to make proper predictions.

3.1 Optimization of the used models

The results tables show the number of iteration cycles needed to reach the target of pH 6 (±0.2) and pH 7 (±0.2) for the buffer systems ammonium-acetate (am-ac) 1:1 and 1:2, respectively. The reason for different pH targets is the lack of experimental raw data around pH 6 in case of the buffer systems am-ac 1:1 and am-ac 1:2.

Each result for every buffer system is already an average consisting of ten single experiments. The column average represents an average of the average for the according hyperparameter.

For the optimization procedure a dataset-split of 5:95% was used. That means 5% of the available data for each of the 18 mixtures served as training data and 95% served as test or verification data. The low amount of only 5% training data (3-6 datapoints in this case) prevents the system from getting overfitted.

For detailed results of the single experiments, see appendix B.

3.1.1 Optimization of the Gaussian process

Table 2 provides an overview of the different kernels applied to seven buffer systems. The result of the optimization process is shown in Figures 18,19 and in Table 3.

Table 2: Overview of different kernels used during the optimization process of gauss process regression. RBF stands for radial basis function and C is the hyperparameter for support vector machines (SVM).

K1	C(1.0)*(RBF() + DotProduct())
K2	C(0.1,(1e-5, 1e2))*RBF(100,(1e-3, 1e5))+RBF(12,(1e-3, 1e5))+RBF(1,(1e-3, 1e3))
K3	C(0.1,(1e-5, 1e2))*RBF(1,(1e-3, 1e3))
K4	C(1)*RBF(1,(1e-3, 1e3))
K5	C(1)*RBF(1,(1e-1, 1e3))

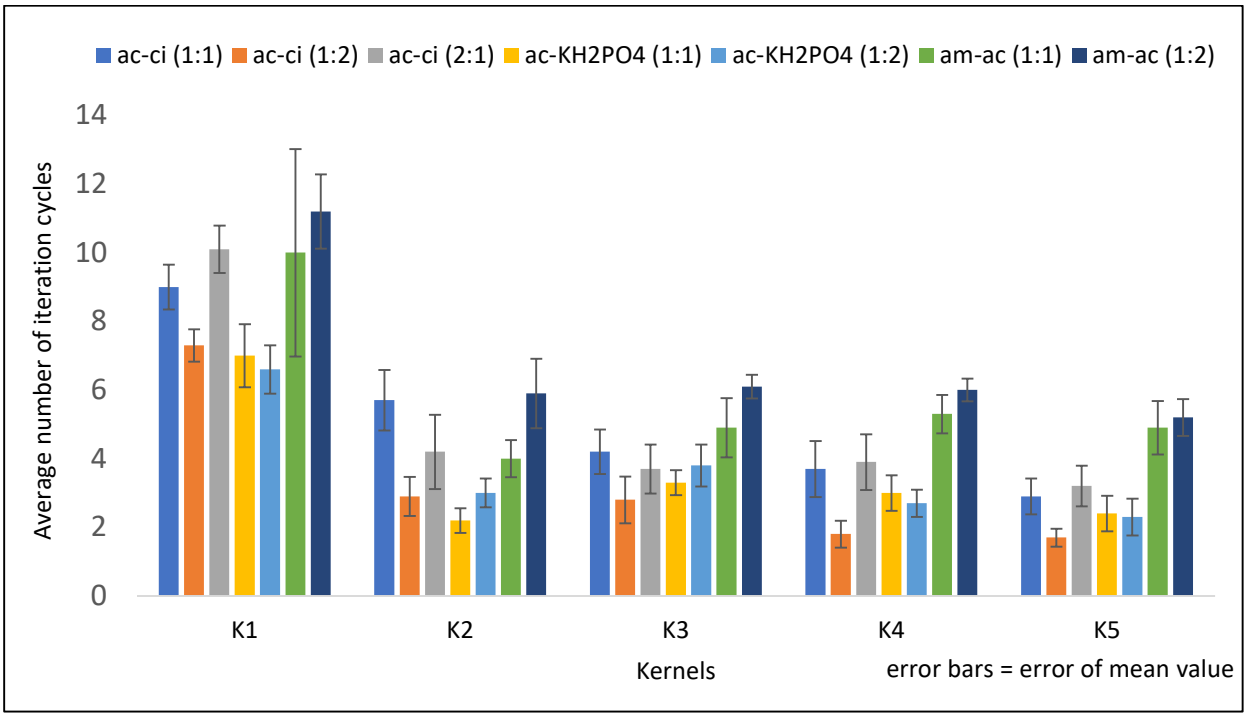

Figure 18: Comparison of different kernels for the optimization of the gaussian process regression (GPR). Each bar represents the calculated average of ten single experiments. The error bars represent the error of mean value.

Table 3: Results of different kernels for the optimization of the gaussian process regression (GPR). The numbers below the binary buffer systems represent an average of ten single experiments. The "average" column is the arithmetic mean for each learning rate and SEM is the standard error of the mean.

	ac-ci (1:1)	ac-ci (1:2)	ac-ci (2:1)	ac- KH_2PO_4 (1:1)	ac- KH_2PO_4 (1:2)	am-ac (1:1)	am-ac (1:2)	average	SEM
K1	9.0	7.3	10.1	7.0	6.6	10.0	11.2	8.74	0.63
K2	5.7	2.9	4.2	2.2	3.0	4.0	5.9	3.99	0.49
K3	4.2	2.8	3.7	3.3	3.8	4.9	6.1	4.11	0.38
K4	3.7	1.8	3.9	3.0	2.7	5.3	6.0	3.77	0.51
K5	2.9	1.7	3.2	2.4	2.3	4.9	5.2	3.23	0.47

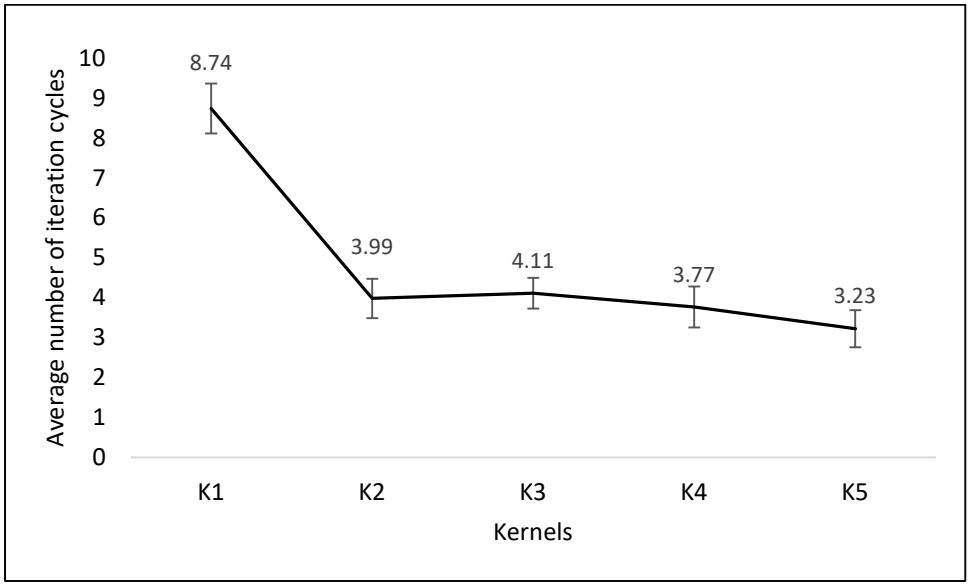

Figure 19: Average number of iteration cycles of 10 single titration experiments. Five different kernels were tested for their ability to reach a pre-defined pH target. Error bars representing the error of the mean value.

The kernels 1-5 are commonly used ones. Each of them has their strengths and weaknesses in different tasks. The dot product portion of kernel 1 (K1) clearly shows a disadvantage when it comes to achieving a low number of iteration cycles.

There is hardly any noticeable difference in average iteration cycles between kernels 2 to 5. Kernel 5 was chosen for the following benchmark.

3.1.2 Optimization of the artificial neural network

The ANN was optimized by manually changing the values of each single hyperparameter. These are learning rate, epochs, activation function, number of neurons and number of layers. Table 4 shows the final hyperparameters after optimization.

Table 4: Final set of hyperparameters of the artificial neural network (ANN) resulting from a manual optimization.

Learning rate	0.015
Epochs	1000
Activation function	ELU
Number of neurons	40
Layers	3

3.1.2.1 *Learning rate*

The learning rate determines the step size during each iteration towards the target. If the learning rate is too small, the number of iterations and thus the calculation time will be disproportionately high. On the other hand, if the learning rate is too high, the method can fail to converge at all (Figure 20) [60].

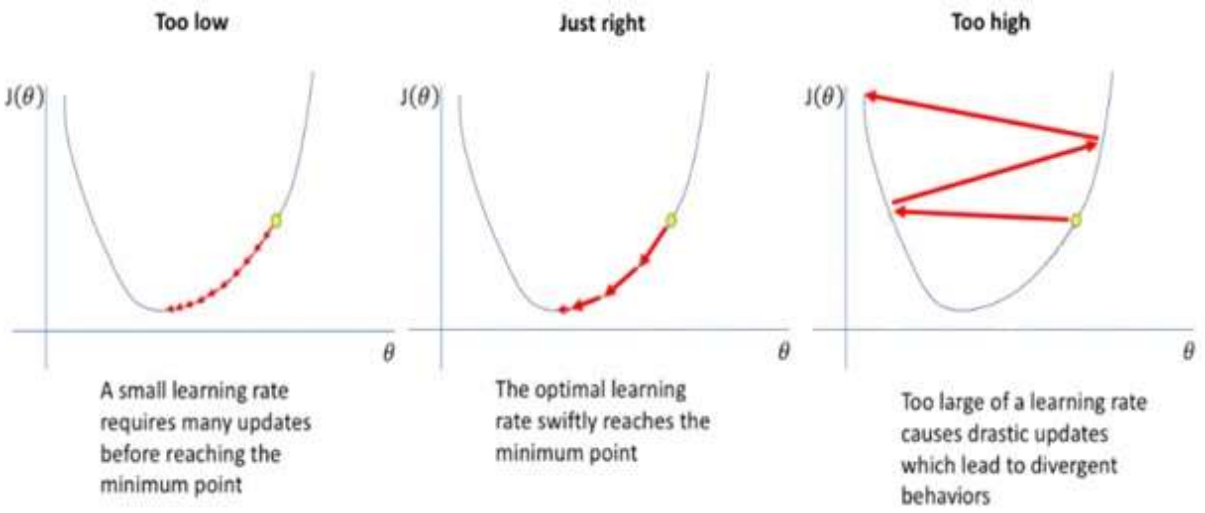

Figure 20: Graphic representation of the learning rate. If the learning rate is too low, the iteration process takes too long and is inefficient (left). If the learning rate is too high, the problem of overshooting increasingly occurs (right). The middle representation shows an idealized approximation of the target with a proper learning rate. [61]

The lower the number of average iterations, the more efficient the system. The minimum number of cycles was determined by starting with a learning rate of 0.005 and was increased by 0.01 increments towards a learning rate of 0.035. The results are represented by Figures 21, 22 and Table 5.

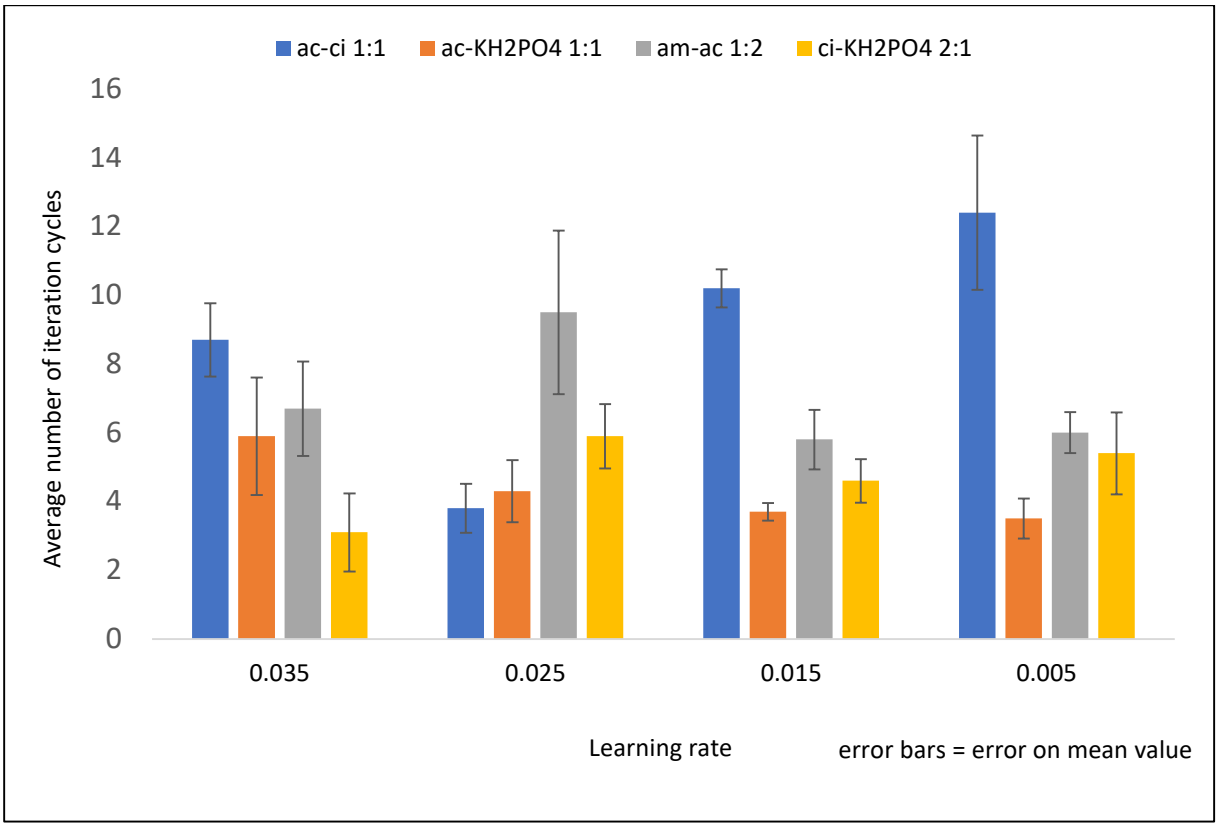

Figure 21: Comparison of different learning rates for the optimization of the artificial neural network (ANN). Each bar represents the calculated average of ten single experiments. The error bars represent the error of mean value.

Table 5: Result of different learning rates for the optimization of the artificial neural network (ANN). The numbers below the binary buffer systems represent an average of ten single experiments. The "average" column is the arithmetic mean for each learning rate and SEM is the standard error of the mean.

Learning rate	ac-ci (1:1)	ac-KH_2PO_4 (1:1)	am-ac (1:2)	ci-KH_2PO_4 (2:1)	average	SEM
0.035	8.7	5.9	6.7	3.1	6.10	1.00
0.025	3.8	4.3	9.5	5.9	5.87	1.12
0.015	10.2	3.7	5.8	4.6	6.07	1.25
0.005	12.4	3.5	6.0	5.4	6.83	1.67

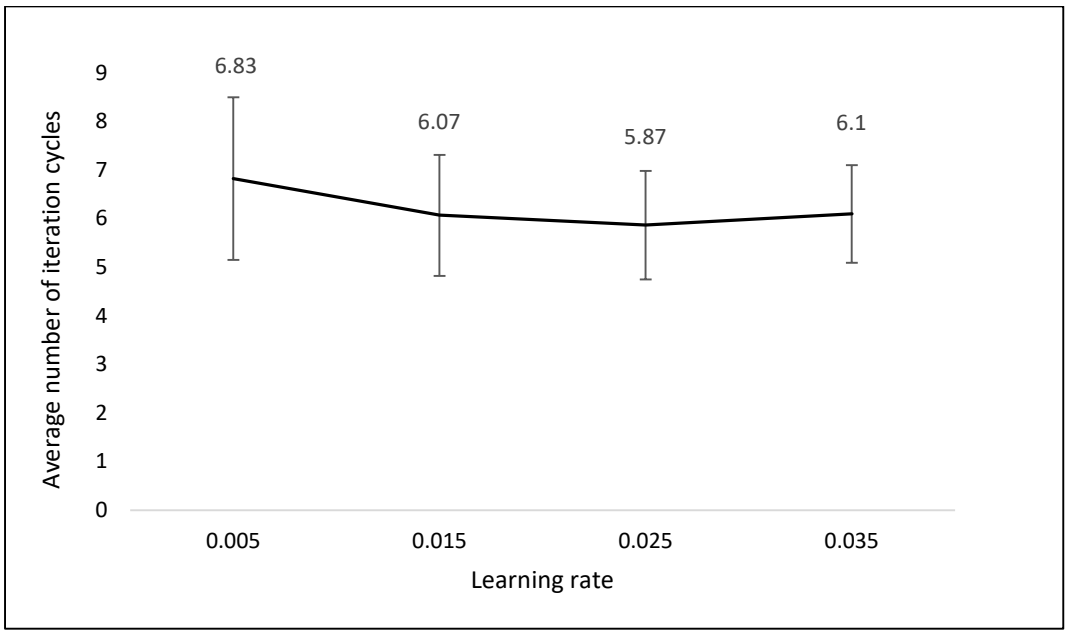

Figure 22: Average number of iteration cycles of 10 single titration experiments. Four different learning rate values were tested for their ability to reach a pre-defined ph target. Error bars representing the error of the mean value.

A learning rate of 0.005 is clearly too low. Learning rates between 0.015 and 0.035 are inside a small range of 4%. Although a learning rate of 0.025 had the least average iterations, we decided to continue with a learning rate of 0.015 because of smaller standard deviation in favor of robustness. In addition to that, the 0.015 learning rate

showed better results when combining them with the other optimized hyperparameters of the ANN.

3.1.2.2 Epochs

One epoch represents one full training cycle of the ANN. All available data will exactly be used once. [62]

It is self-explanatory, that the computing time increases with a higher number of epochs. For this reason, the target is to find the lowest possible number of epochs with reasonable accuracy. Accuracy means low number of iteration cycles in this case. Figures 23, 24 and Table 6 are representing the results of the optimization process.

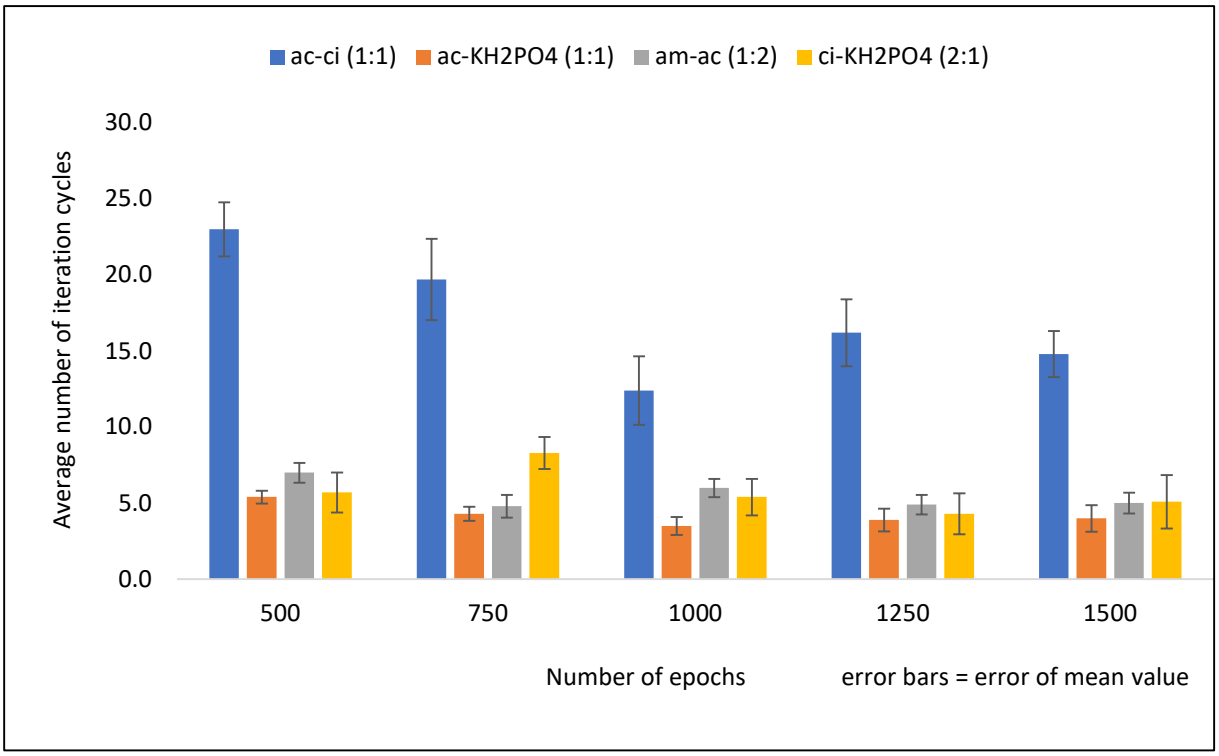

Figure 23: Comparison of different epoch values for the optimization of the artificial neural network (ANN). Each bar represents the calculated average of ten single experiments. The error bars represent the error of mean value.

Table 6: Result of different epoch values for the optimization of the artificial neural network (ANN). The numbers below the binary buffer systems represent an average of ten single experiments. The "average" column is the arithmetic mean for each learning rate and SEM is the standard error of the mean.

Epochs	ac-ci (1:1)	ac-KH_2PO_4 (1:1)	am-ac (1:2)	ci-KH_2PO_4 (2:1)	average	SEM
500	23.0	5.4	7.0	5.7	10.28	3.69
750	19.7	4.3	4.8	8.3	9.28	3.11
1000	12.4	3.5	6.0	5.4	6.83	1.67
1250	16.2	3.9	4.9	4.3	7.33	2.57
1500	14.8	4.0	5.0	5.1	7.23	2.20

On the first view it looks counterintuitive to plot the average number of iteration cycles because of the huge discrepancy between the single binary buffer systems. Specially in this case it could be considered to print the sum of the average number of iteration cycles instead of the average of it. For consistency reasons it was decided to use them anyway.

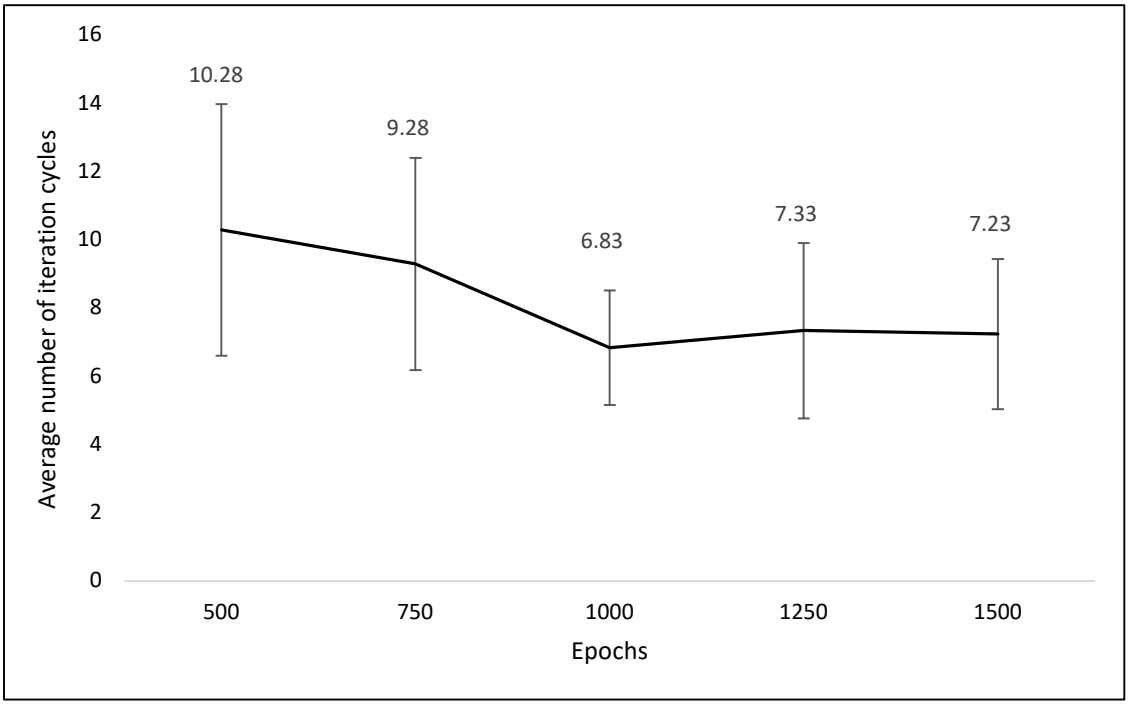

Figure 24: Average number of iteration cycles of 10 single titration experiments. Five different numbers of epochs were tested for their ability to reach a pre-defined pH target. Error bars representing the error of the mean value.

A slight drop of iteration cycles occurred between 500 and 1000 epochs. A value of 1250 and 1500 epochs respectively showed a slight increase in average iteration cycles. This value is also acceptable in terms of computational time. Therefore 1000 epochs gave the best result.

3.1.2.3 Activation function

In artificial neural networks typically each neuron in the hidden layer has a non-linear activation function (Figure 25).

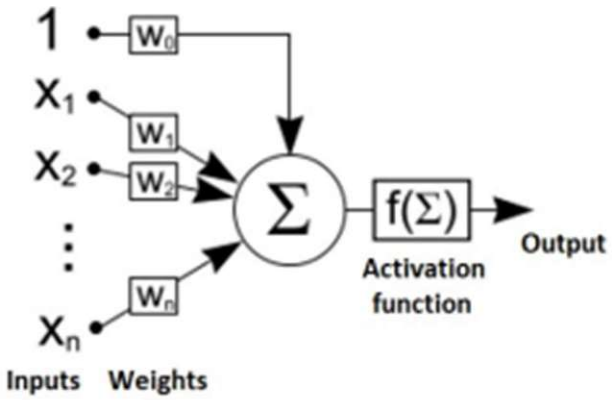

Figure 25: Illustration of output calculation in artificial neural networks (ANN) using activation functions [63]

The activation function is responsible for generating non-linearity out of a former linear system. Several pre-defined activation functions are available. The most common used ones are sigmoid function, hyperbolic tangent function (Tanh), rectified linear unit function (ReLU), and exponential linear unit (ELU) [63], [64]. A detailed description of the activation functions mentioned can be found in the appendix B. Figure 26, 27 and Table 7 showing the results of the optimization process.

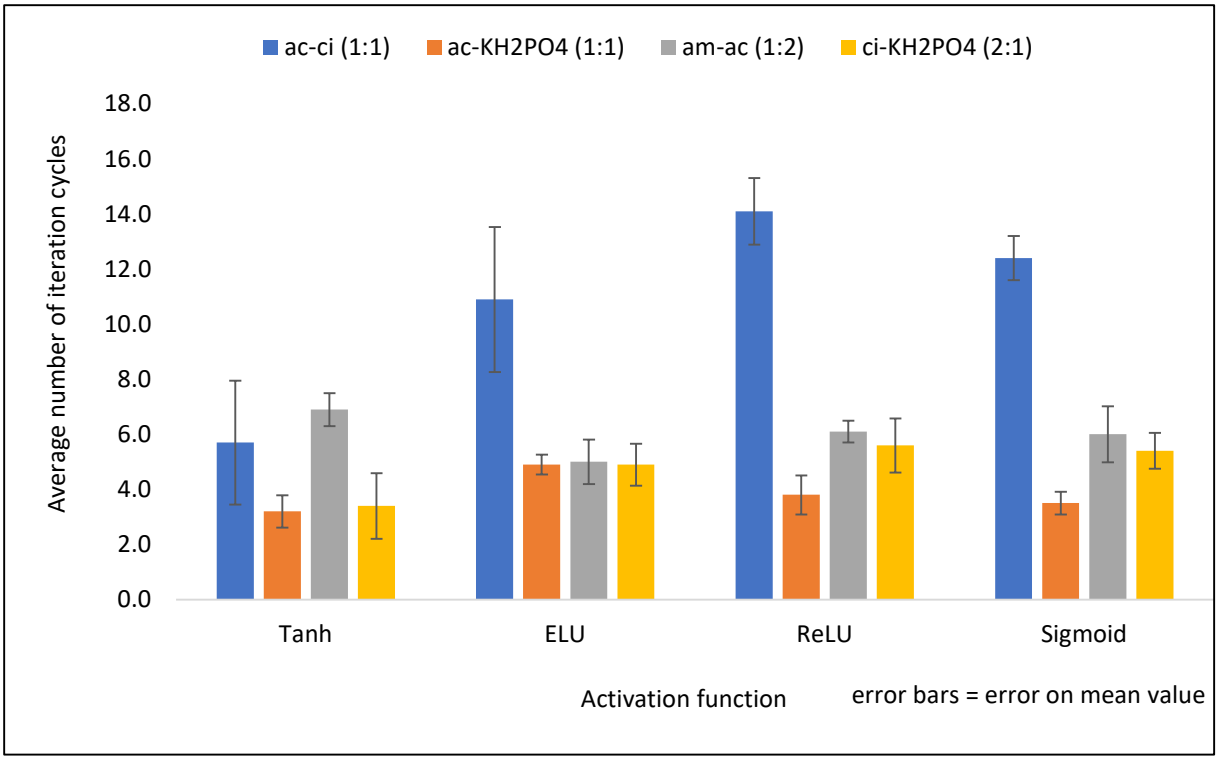

Figure 26: Comparison of different activation functions for the optimization of the artificial neural network (ANN). Each bar represents the calculated average of ten single experiments. The error bars represent the error of mean value.

Table 7: Result of different activation functions for the optimization of the artificial neural network (ANN). The numbers below the binary buffer systems represent an average of ten single experiments. The "average" column is the arithmetic mean for each learning rate and SEM is the standard error of the mean.

Activation function	ac-ci (1:1)	ac-KH_2PO_4 (1:1)	am-ac (1:2)	ci-KH_2PO_4 (2:1)	average	SEM
Tanh	5.7	3.2	6.9	3.4	4.80	0.78
ELU	10.9	4.9	5.0	4.9	6.43	1.29
ReLU	14.1	3.8	6.1	5.6	7.40	1.98
Sigmoid	12.4	3.5	6.0	5.4	6.83	1.67

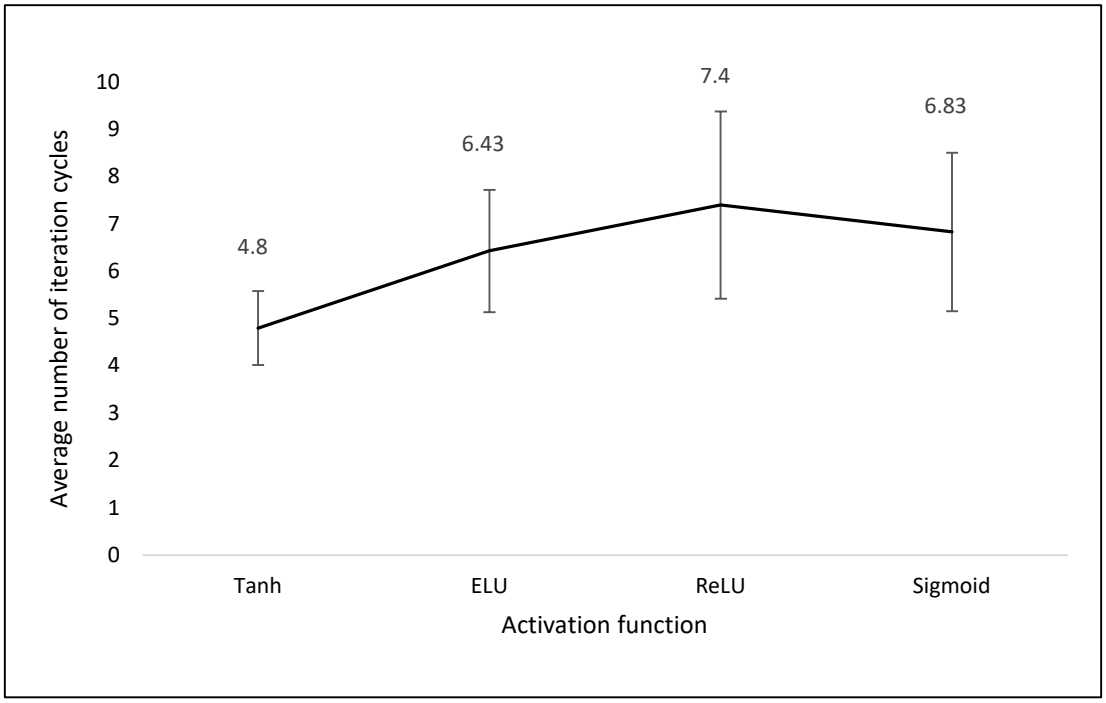

Figure 27: Average number of iteration cycles of 10 single titration experiments. Four different activation functions were tested for their ability to reach a pre-defined pH target. Error bars representing the error of the mean value.

Apparently, the activation function Tanh gave the best results. However, it performed very badly during the benchmark in combination with the other optimized parameters. Therefore, the second-best activation function, ELU was tested with the optimized algorithm, and it performed significantly better. This shows the problem of empirical hyperparameter optimization. However, this approach is sufficient to compare the methods presented. Alternatively, there are a lot of different software guided optimization techniques. M. Abdolrasol *et al.* showed different commonly used strategies for further information. [65]

3.1.2.4 Number of neurons

Like in human brain, neurons in artificial neuronal networks are responsible to process information and hand them over to the directly connected neurons, if it receives a sufficiently strong input signal (Figure 28). [66]

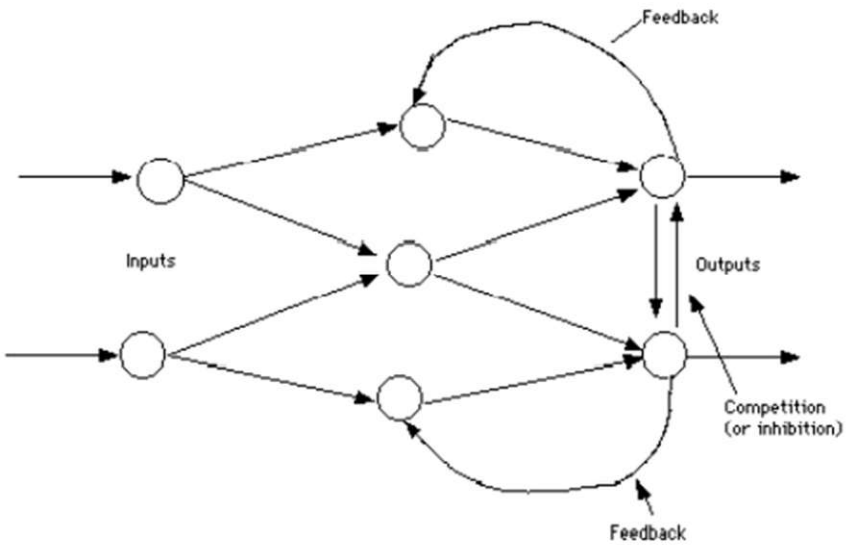

Figure 28: Visualization of a simple artificial neuronal network (ANN) with feedback loop [66]

Increasing the number of neurons in a single hidden layer typically leads to a decreasing mean squared error but increases the computational complexity on the other hand [67]. Figures 29, 30 and Table 8 showing the results of the optimization process.

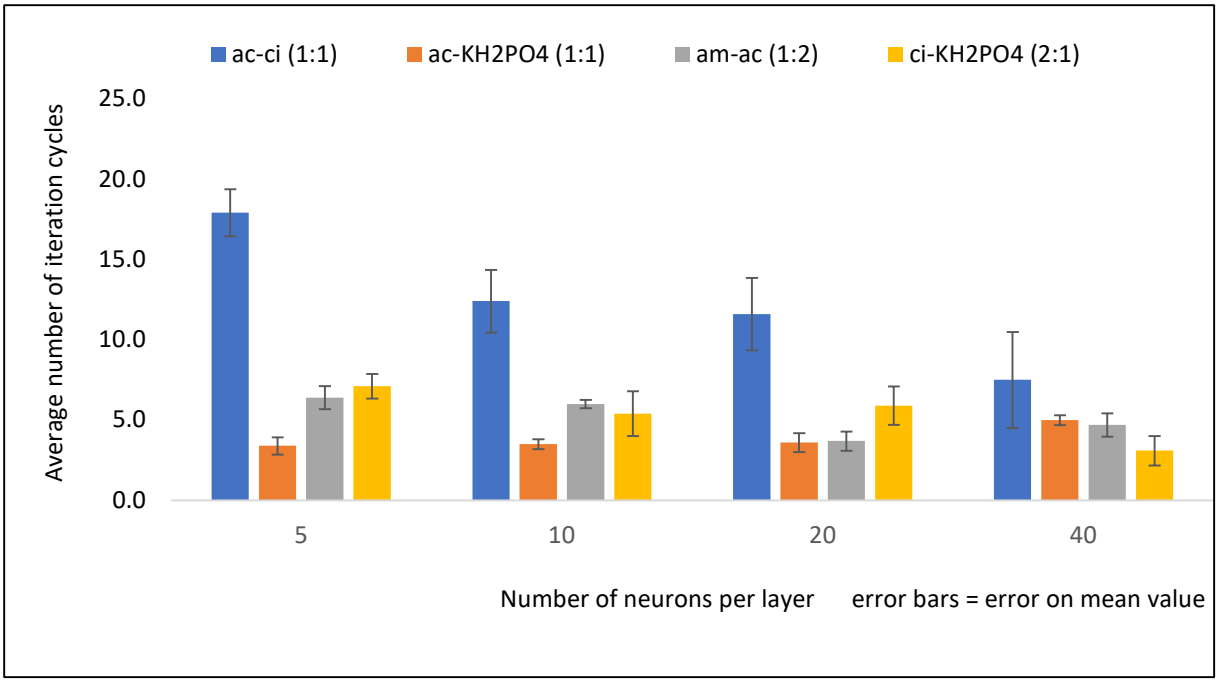

Figure 29: Comparison of different neuron numbers per layer for the optimization of the artificial neural network (ANN). Each bar represents the calculated average of ten single experiments. The error bars represent the error of mean value.

Table 8: Result of different number of neurons for the optimization of the artificial neural network (ANN). The numbers below the binary buffer systems represent an average of ten single experiments. The "average" column is the arithmetic mean for each learning rate and SEM is the standard error of the mean.

Number of Neurons	ac-ci (1:1)	ac-KH_2PO_4 (1:1)	am-ac (1:2)	ci-KH_2PO_4 (2:1)	average	SEM
5	17.9	3.4	6.4	7.1	8.70	2.75
10	12.4	3.5	6.0	5.4	6.83	1.67
20	11.6	3.6	3.7	5.9	6.20	1.63
40	7.5	5.0	4.7	3.1	5.08	0.79

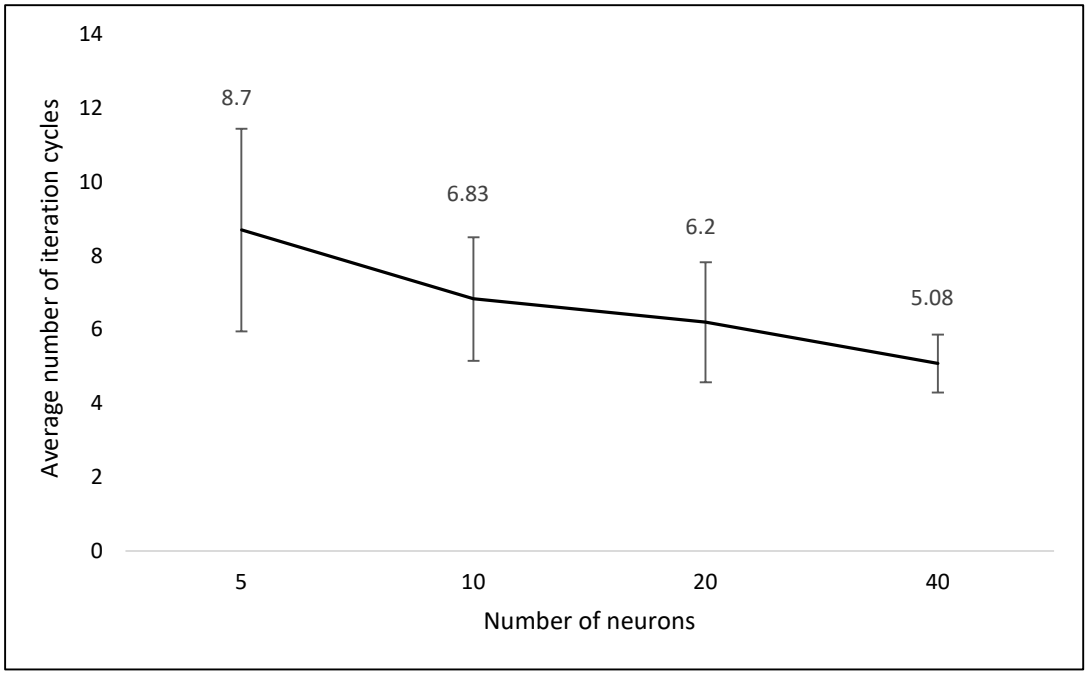

Figure 30: Average number of iteration cycles of 10 single titration experiments. Four different numbers of neurons were tested for their ability to reach a pre-defined pH target. Error bars representing the error of the mean value.

There is a continuous decrease of average iteration cycles when increasing the number of neurons per layer. Although the buffer system acetate-KH_2PO_4 does not reflect this trend, the average number of iteration cycles clearly negatively correlates with a higher number of neurons. More neurons mean more computational time though. With respect to calculation time per iteration the experiment was stopped at 40 neurons per layer. This value gave a reasonable accuracy : time ratio.

3.1.2.5 Hidden layers

Neurons are arranged in a so-called layer. One differentiates between input-, hidden- and output layers (Figure 31).

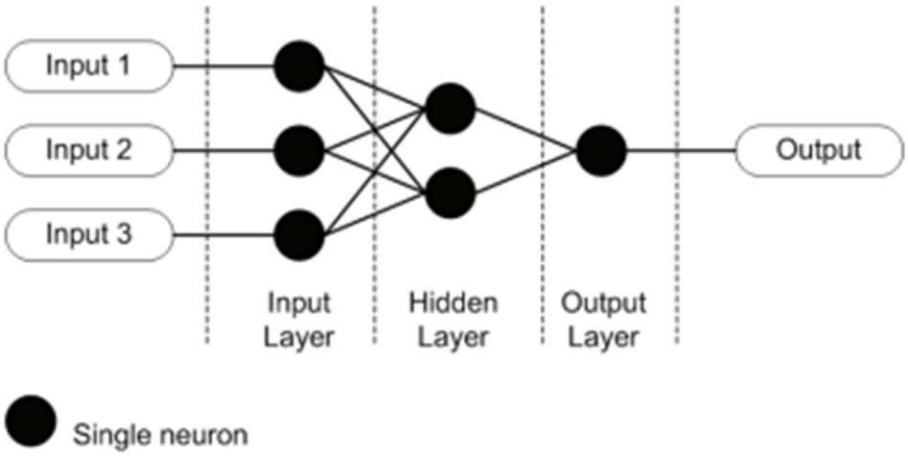

Figure 31: Simplified artificial neural network (ANN) diagram [68]

Increasing the number of hidden layers in the network typically leads to higher accuracy, but increased calculation time on the other hand. [69] Figures 32, 33 and Table 9 showing the results of the optimization process.

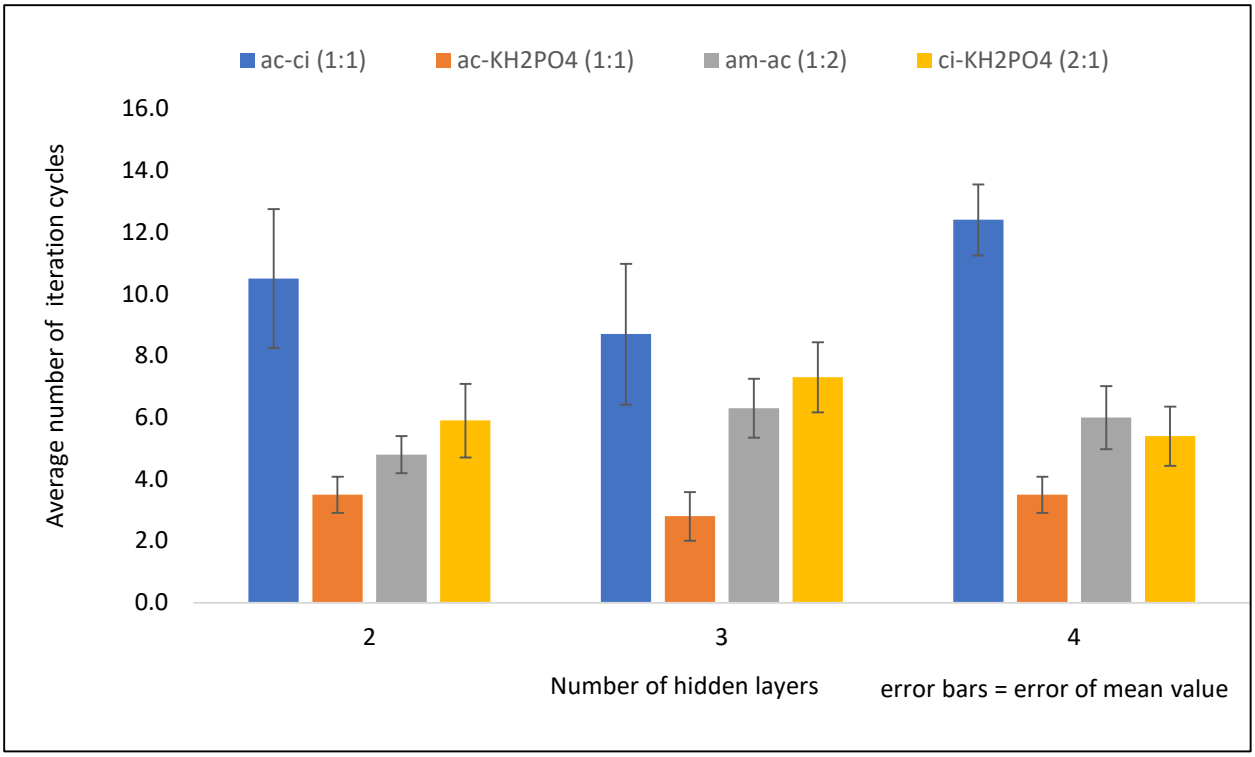

Figure 32: Comparison of different number of hidden layers for the optimization of the artificial neural network (ANN). Each bar represents the calculated average of ten single experiments. The error bars represent the error of mean value.

Table 9: Result of different number of hidden layers for the optimization of the artificial neural network (ANN). The numbers below the binary buffer systems represent an average of ten single experiments. The "average" column is the arithmetic mean for each learning rate and SEM is the standard error of the mean.

Hidden layers	ac-ci (1:1)	ac-KH_2PO_4 (1:1)	am-ac (1:2)	ci-KH_2PO_4 (2:1)	average	SEM
2	10.5	3.5	4.8	5.9	6.18	1.32
3	8.7	2.8	6.3	7.3	6.28	1.09
4	12.4	3.5	6.0	5.4	6.83	1.67

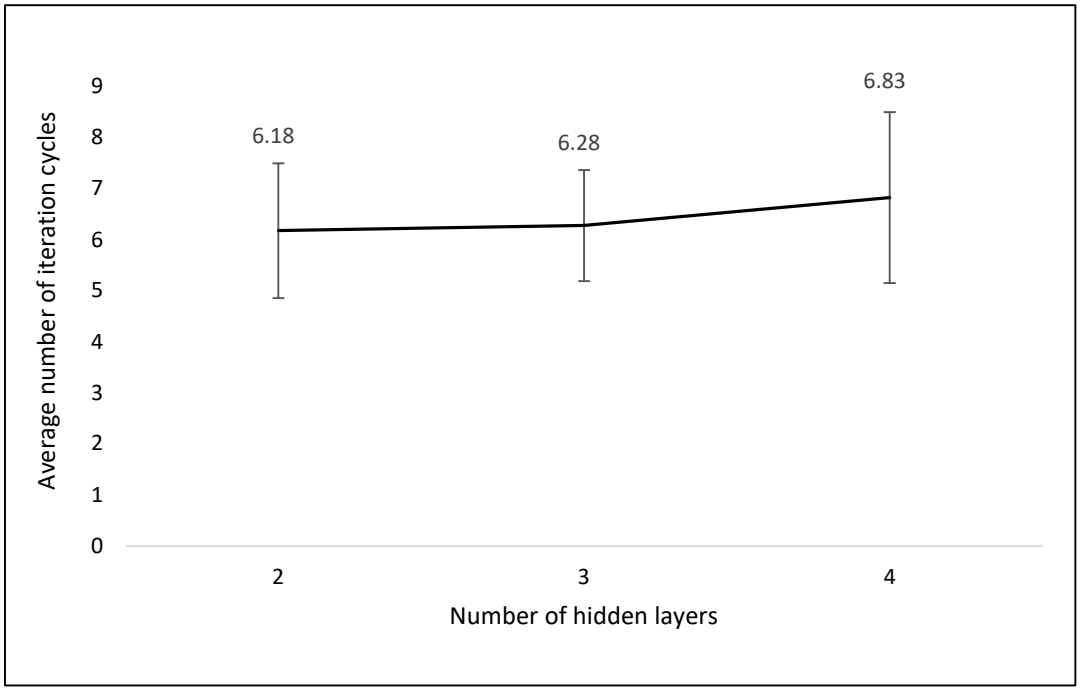

Figure 33: Average number of iteration cycles of 10 single titration experiments. Three different numbers of hidden layers were tested for their ability to reach a pre-defined pH target. Error bars representing the error of the mean value.

The optimization of hidden layers was performed with 10 neurons per layer as a standard value. It shows a tendency but no significance of higher numbers of average iteration cycles the more hidden layers were used. According to the results, it seems that 2 hidden layers gave the best result. But in connection with the optimized value for the number of neurons per layer (40 neurons) the accuracy decreased significantly. For this reason, we decided to increase to 3 hidden layers, which gave good results. Figure 33 clearly shows that the number of hidden layers has very little influence on the performance of the pH adjustment process.

As already stated for the optimization process of the activation function, the empirical search for optimal hyperparameters has its weakness, but this approach is sufficient to compare the methods presented.

3.1.3 Optimization of the random forest

The only hyperparameter for random forest models is the number of estimators and it represents the number of decision trees in the forest. Figures 34, 35 and Table 10 show the results of the optimization process.

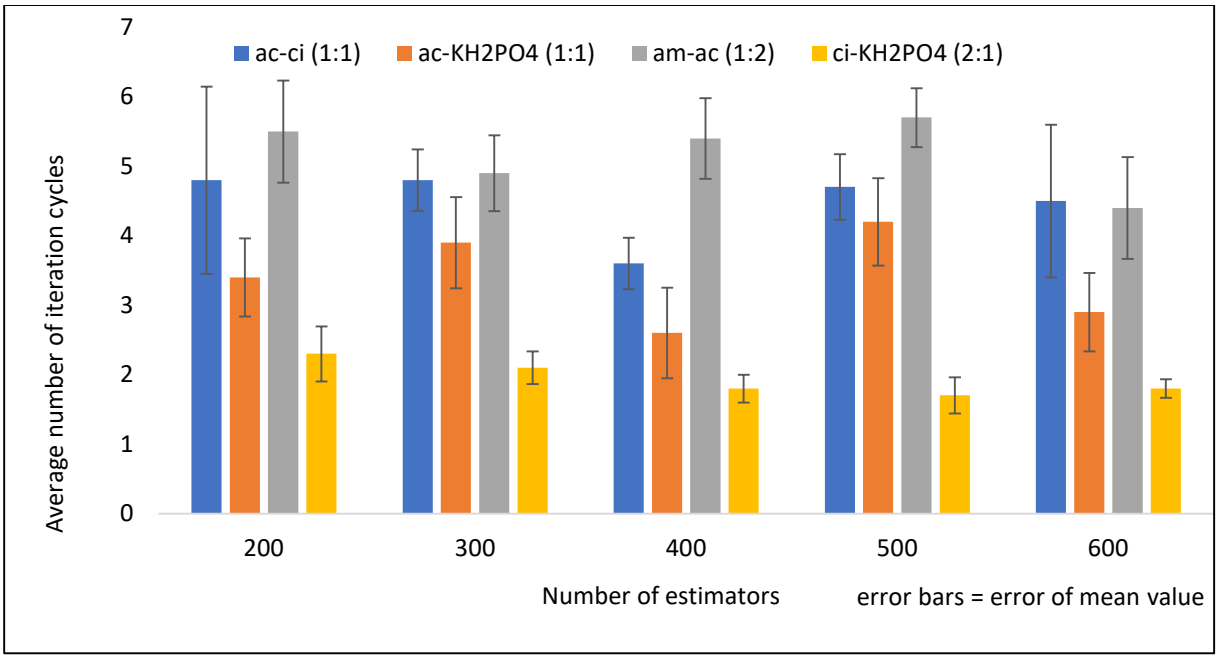

Figure 34: Comparison of different number of estimators for the optimization of the random forest (RF). Each bar represents the calculated average of ten single experiments. The error bars represent the error of mean value.

Table 10: Result of different number of estimators for the optimization of the random forest (RF). The numbers below the binary buffer systems represent an average of ten single experiments. The "average" column is the arithmetic mean for each learning rate and SEM is the standard error of the mean.

Estimators	ac-ci (1:1)	ac-KH_2PO_4 (1:1)	am-ac (1:2)	ci-KH_2PO_4 (2:1)	average	SEM
200	4.8	3.4	5.5	2.3	4.00	0.62
300	4.8	3.9	4.9	2.1	3.93	0.56
400	3.6	2.6	5.4	1.8	3.35	0.67
500	4.7	4.2	5.7	1.7	4.08	0.74
600	4.5	2.9	4.4	1.8	3.40	0.56

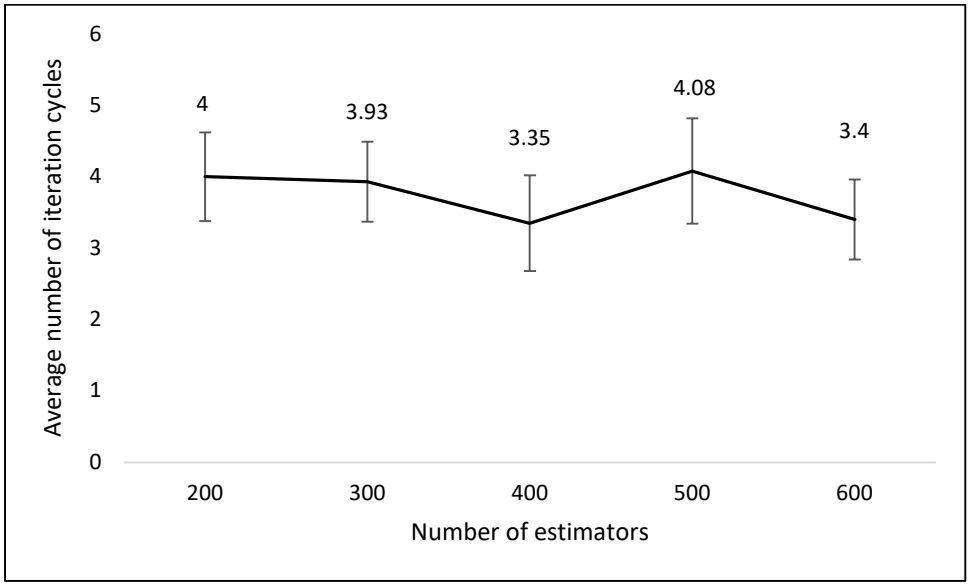

Figure 35: Average number of iteration cycles of 10 single titration experiments. Five different numbers of estimators were tested for their ability to reach a pre-defined pH target. Error bars representing the error of the mean value.

A wide range of different estimator values was tested. The results just moved within a 21% range. It seems that the number of estimators is not that critical for the current task. However, the number of 400 estimators showed a local minimum at an average of 3.35 iteration cycles to predict the correct target. With respect to calculation time on

an average modern personal computer, 400 estimators were chosen as the random forest hyperparameter.

3.2 Comparison of the single benchmarks

Figure 36 and Table 11 showing a direct comparison of the benchmark between artificial neural network (ANN), random forest (RF), linear regression and gaussian process regression (GPR).

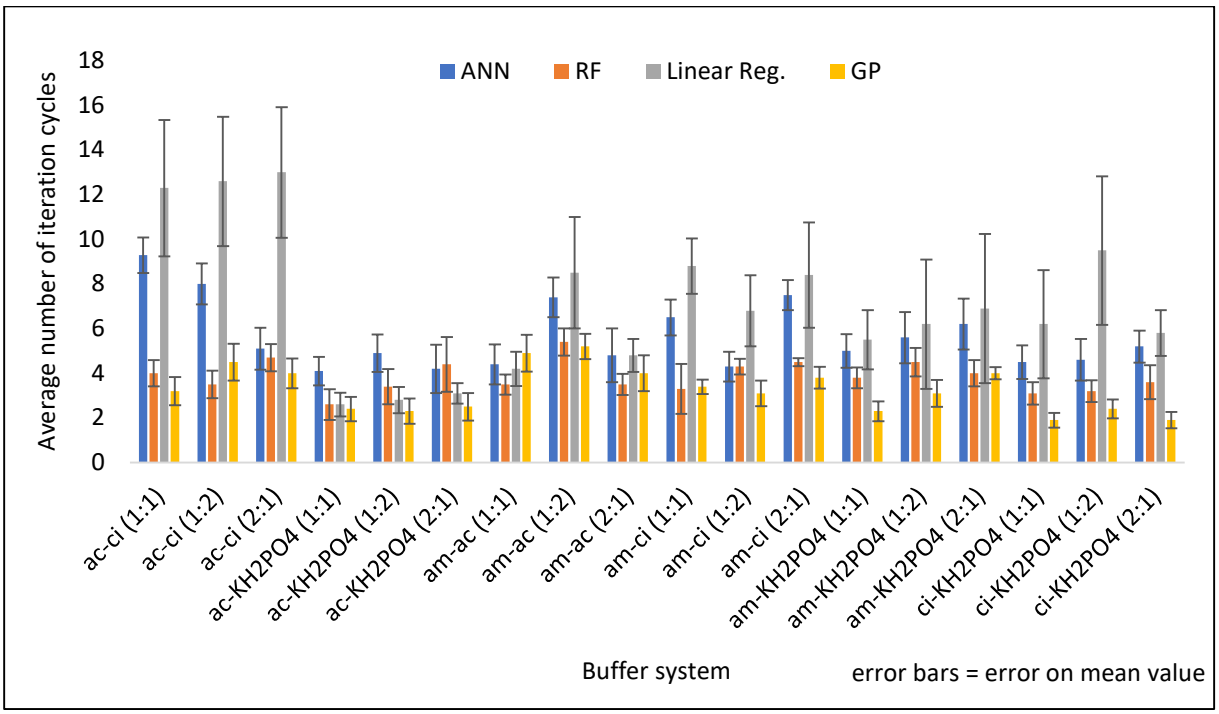

Figure 36: Direct comparison of the benchmark result for the artificial neural network (ANN), random forest (RF), linear regression and gaussian process regression (GPR). Each bar represents the calculated average of ten single experiments. The error bars represent the error of mean value.

Table 11: Benchmark result of average number of iteration cycles needed to reach the target pH for the artificial neural network (ANN), random forest (RF), linear regression and gaussian process regression (GPR) with the corresponding error on mean values (SEM).

regression method	average iteration cycles (± error on mean value)
ANN	5.6 (±1.0)
RF	3.4 (±0.3)
Linear Reg.	7.1 (±1.9)
GP	3.1 (±0.6)

Linear regression showed a significantly higher average iteration value than other methods for some buffer systems. It performed very poorly, especially with the acetate-citrate buffer system The performance of linear regression is highly dependent on the curve shape. Good results only show up in linear areas of the titration curve.

Artificial neural networks are tricky in regards of optimization because of five different hyperparameters that all influence each other. That makes the hyperparameter optimization a time-consuming task. That is why further investigation of the ANN hyperparameters could have the largest potential to further reduce the average number of iteration cycles of this algorithm.

Random forest performed very well. Beside the low value of average iteration cycles, it showed the smallest error on mean value. In addition to the fact that it is very easy and robust in terms of optimization, it also has low computational demands.

The Gaussian process showed the best results in terms of average iteration cycles. The error on mean value was slightly above random forest. The advantage of GP lies in the quick calculation of the results which means less computational time. This should be considered when it comes to high throughput tasks.

It needs to be mentioned that the results of *all* algorithms could still be improved by further optimization. However, for the purpose of deciding which algorithm will be chosen for further studies, the results are already sufficient.

3.2.1 Benchmark of the artificial neural network

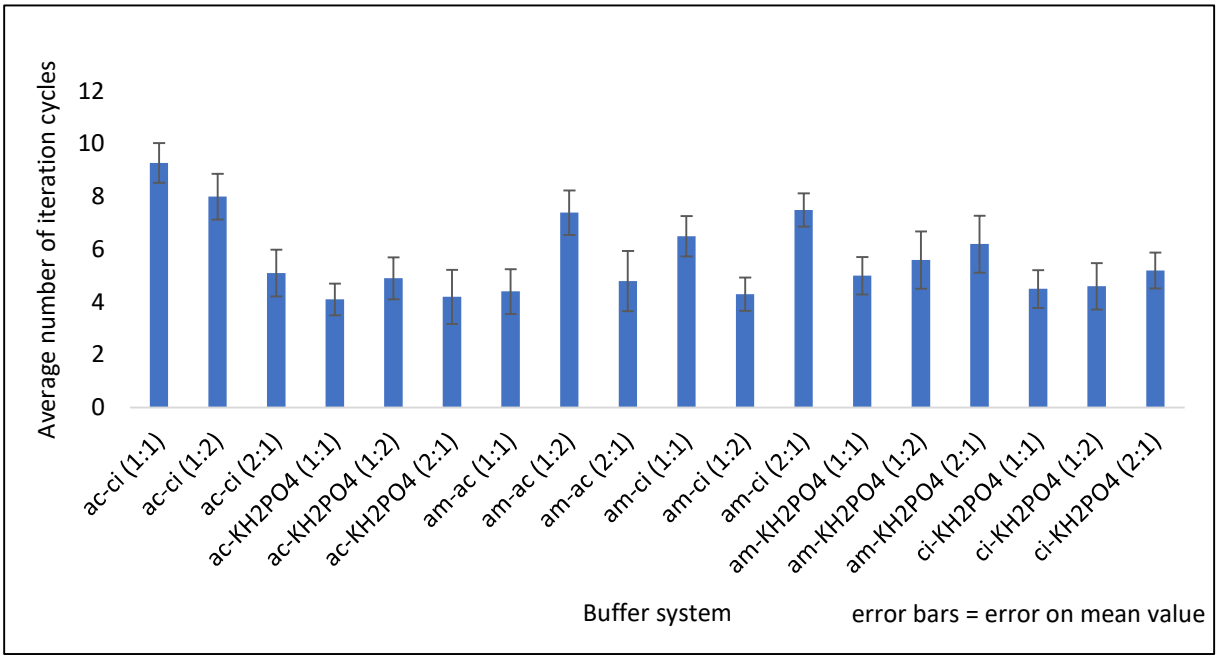

Figure 37: Graphical benchmark results for the artificial neural network (ANN) with optimized hyperparameters. Each bar represents the calculated average of ten single experiments. Error bars representing the error of the mean value.

It took an overall average of 5.64 (±2.63) iteration cycles to reach the target of pH 6 (±0.2) and pH 7 (±0.2) for the buffer systems am-ac 1:1 and 1:2 respectively. Table 4 shows the final hyperparameters and Figure 37 the graphic result of the benchmark after the optimization process. The number of iterations corresponds directly to the shape of the titration curve. If the slope is steep in the target region, it is harder to find the correct values for adjustment. That is because small amounts of acid or base addition will cause big changes in pH value. Each algorithm has a different approach to the task. That is the reason why it is vital to investigate whether one or another algorithm gives the best results.

For detailed experimental results see appendix C.

3.2.2 Benchmark of the random forest

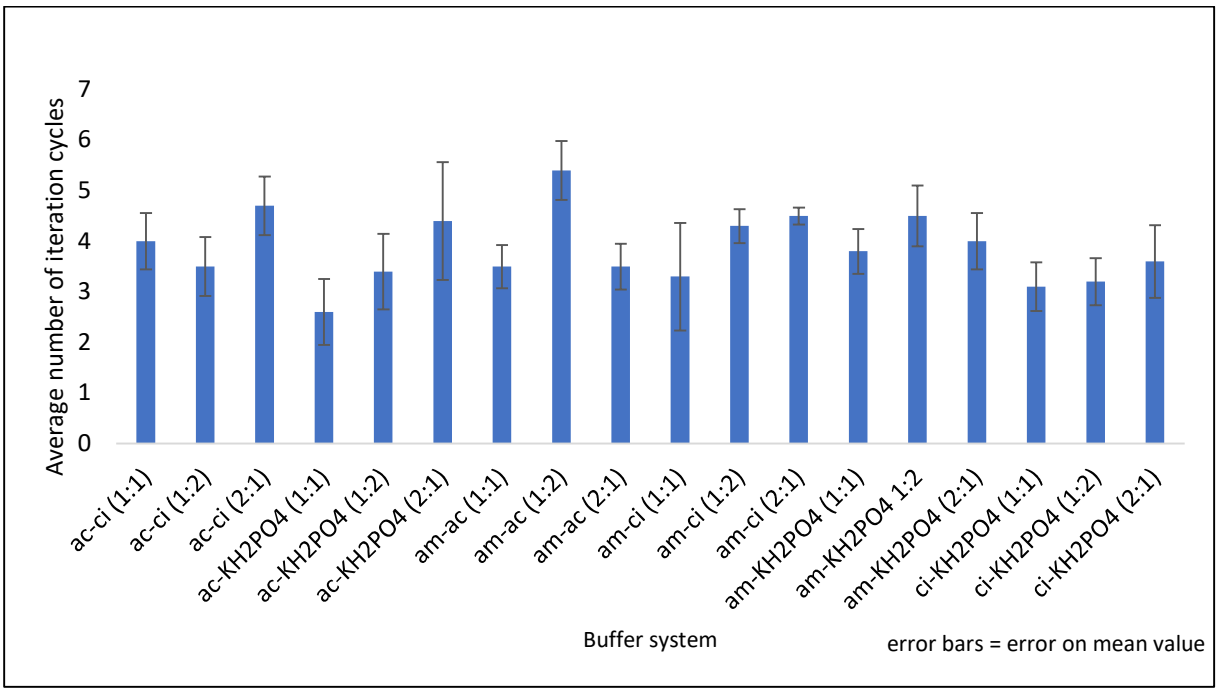

Figure 38: Graphical benchmark results for the random forest (RF) with optimized hyperparameter. Each bar represents the calculated average of ten single experiments. Error bars representing the error of the mean value.

It took an overall average of 3.85 (±1.86) iteration cycles to reach the target of pH 6 (±0.2) and pH 7 (±0.2) for the buffer systems am-ac 1:1 and 1:2 respectively. Figure 38 shows the graphic result of the benchmark after the optimization process. Random forest showed relatively constant averages over all buffer mixtures. It needs to be mentioned, that some of the error bars are relatively big, especially for the acetate-KH_2PO_4 mixtures and the ammonium-acetate 1:1 buffer. This is a weak spot of this algorithm. The reason for that might be the shape of the titration curve and therefore the difficult curve fitting. For detailed experimental results see appendix C.

3.2.3 Benchmark of the linear regression

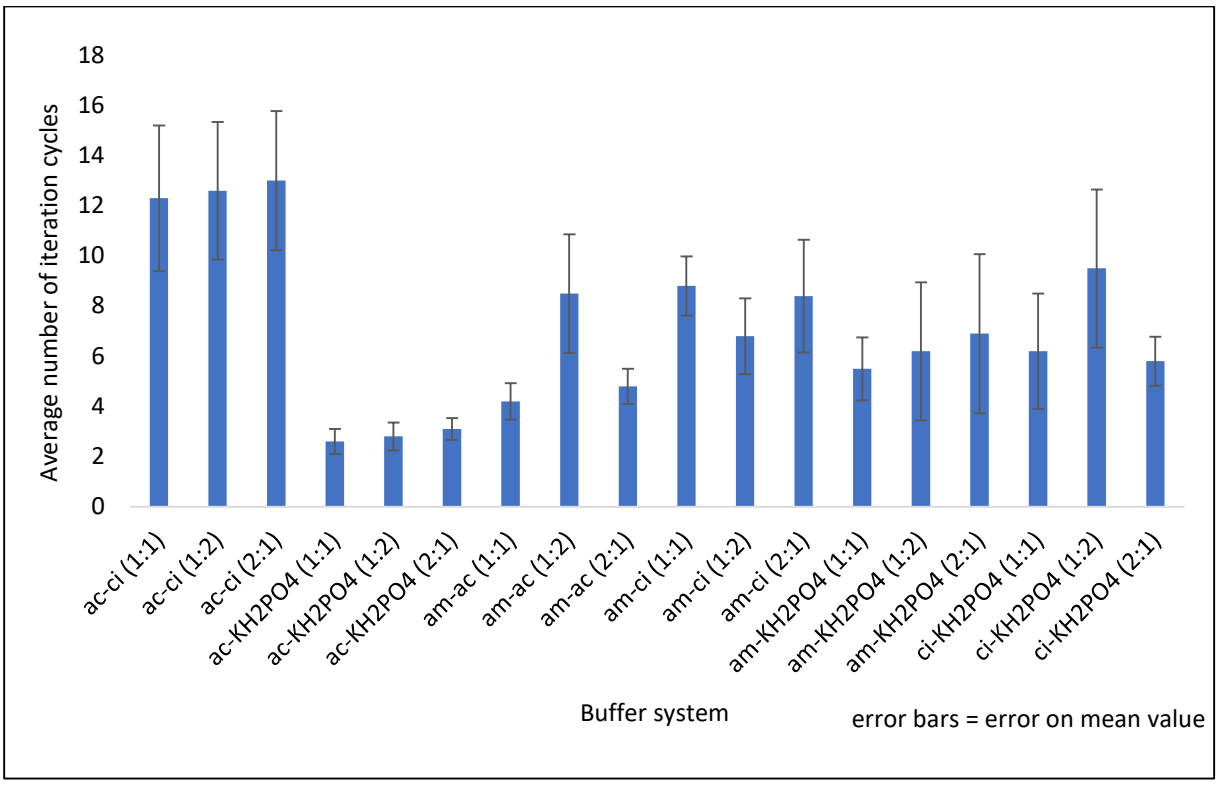

Figure 39: Graphical benchmark results for linear regression. Each bar represents the calculated average of ten single experiments. Error bars representing the error of the mean value.

It took an overall average of 7.11 (±5.67) iteration cycles to reach the target of pH 6 (±0.2) and pH 7 (±0.2) for the buffer systems am-ac (1:1) and (1:2) respectively. Figure 39 show the graphic result of the benchmark. Linear regression performed very well on the acetate-KH_2PO_4 buffers, but poorly on the acetate-citrate buffers. The reason for the poor performance can be explained by a closer look at the titration curves of the buffer systems in Figure 40. Because pH 6 is in a non-linear area of the curve the linear regression method shows a poor performance.

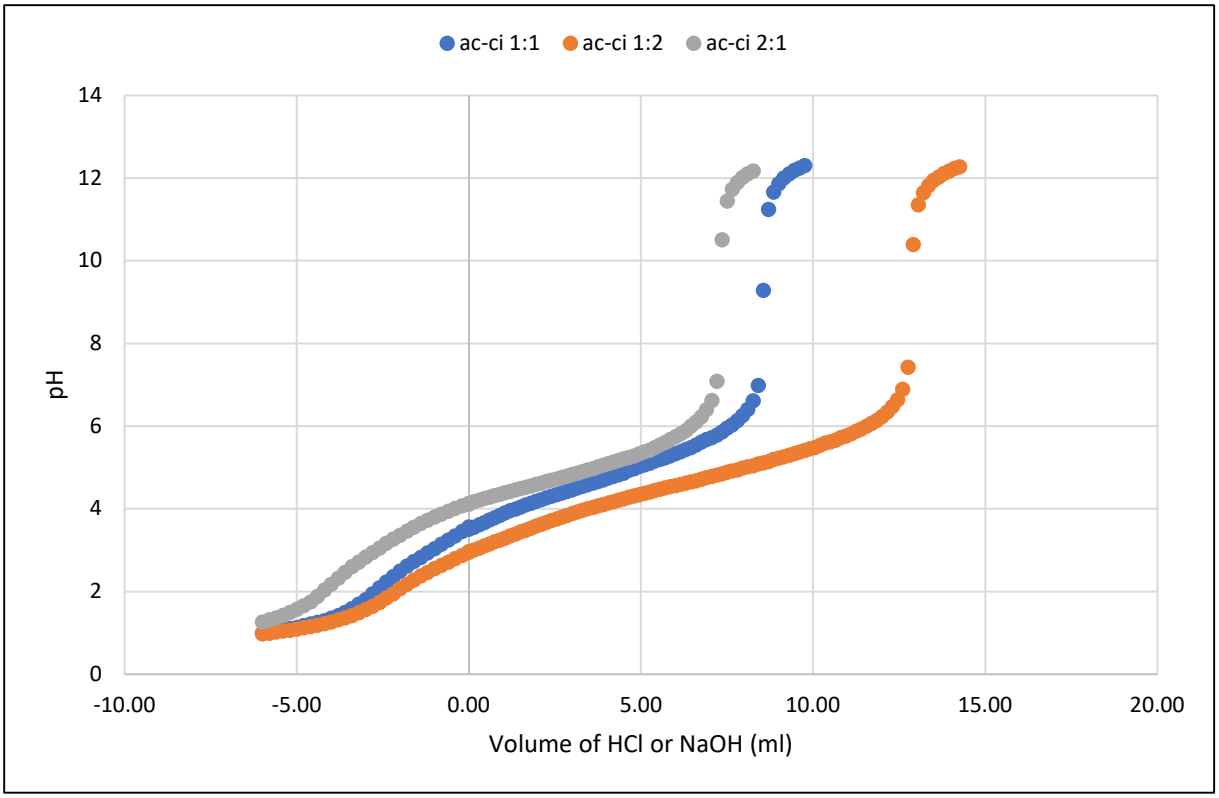

Figure 40: Titration curves for the binary buffer systems acetate-citrate with ratios of 1:1, 1:2 and 2:1. The addition of acid or base is indicated in ml on the x-axis. The addition of acid is indicated by a negative sign and the addition of base by a positive sign.

For the same reason, it is also clear from the titration curve in Figure 41, why linear regression performs so well with the acetate-KH_2PO_4 buffer system. The target pH 6 is located in a linear area of the titration curve, which makes a huge difference in performance of linear regression models.

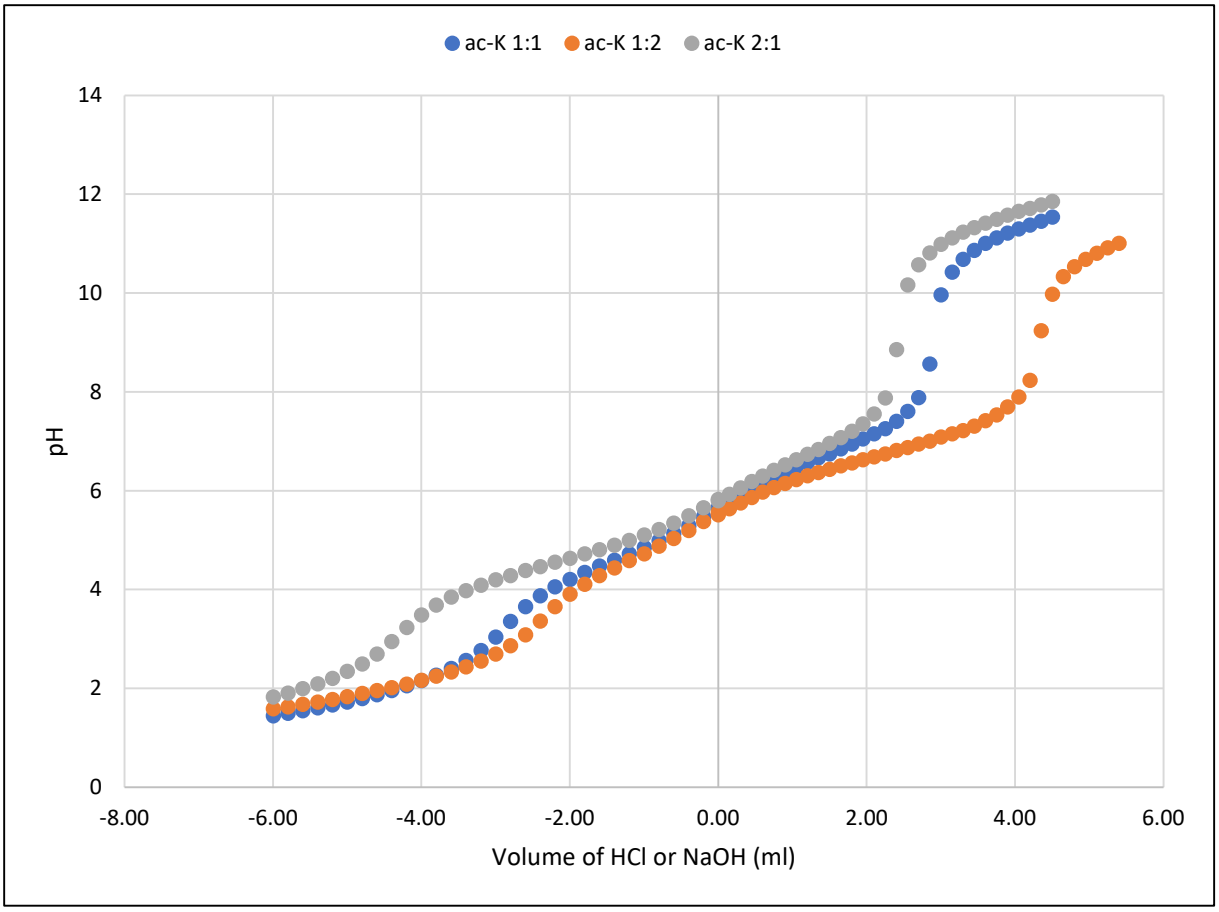

Figure 41: Titration curves for the binary buffer systems acetate-KH_2PO_4 with ratios of 1:1, 1:2 and 2:1. The addition of acid or base is indicated in ml on the x-axis. The addition of acid is indicated by a negative sign and the addition of base by a positive sign.

The error bars are huge for some buffers compared with the other algorithms. That means that the ten single experiments for each bar in the figure differ greatly from one another. For the present task the deviation of the single experiments is less important than the overall average though.

3.2.4 Benchmark of the Gaussian process

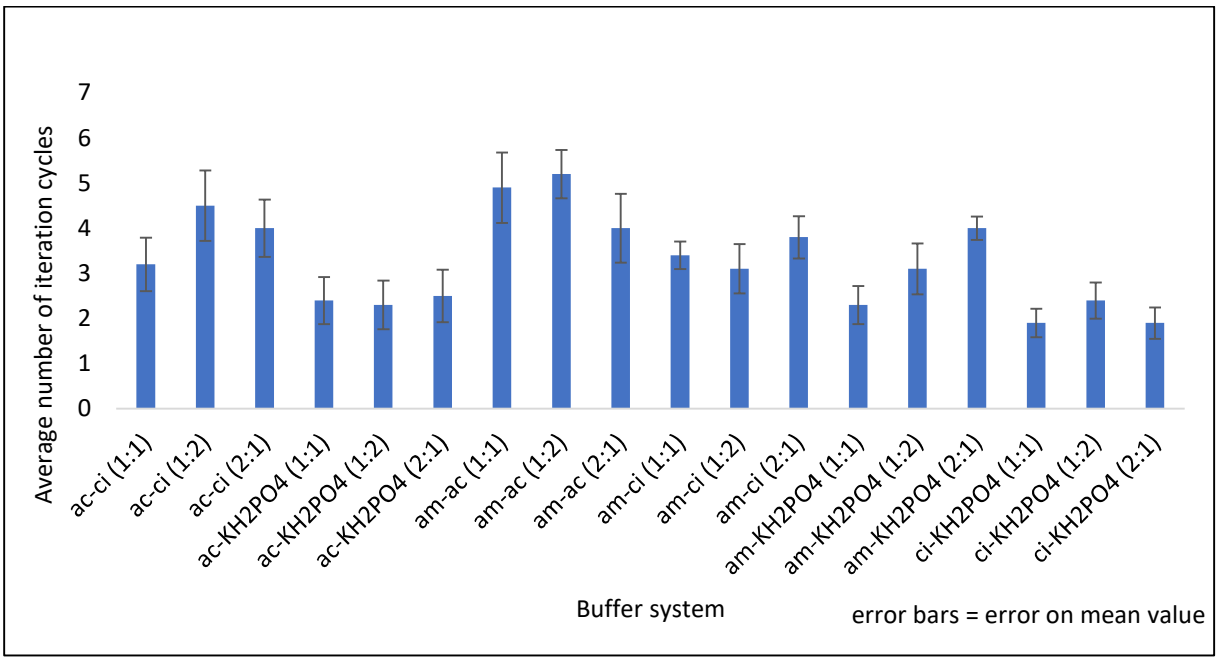

Figure 42: Graphical benchmark results for the gaussian process (GP) with optimized hyperparameter. Error bars represent the error of the mean value.

It took an overall average of 3.27 (±1.64) iteration cycles to reach the target of pH 6 (±0.2) and pH 7 (±0.2) for the buffer systems am-ac 1:1 and 1:2, respectively. Figure 42 shows the graphic result of the benchmark after the optimization process. Gaussian process showed the lowest overall average of iterations and the second lowest error on mean value, beside random forest. As already stated, the overall average value is more important than the value for standard deviation for the present task. We can see a rather even distribution over all buffer mixtures and error bars.

For detailed experimental results see appendix C.

3.3 Feature-set comparison

Features incorporate the chemical information about the used buffer system. In order to find out, whether more chemical information brings an advantage in reducing the number of iteration cycles, the Gaussian process model with optimized hyperparameter was used to compare a small and a large feature-set (Table 12).

Table 12: Parameter overview of different feature sets. The small feature set only provides information about the buffer concentrations, whereas the large feature set additionally includes pKa values, initial pH values, and the number of protons of the buffers.

small feature set	large feature set
concentration buffer 1	concentration buffer 1
concentration buffer 2	concentration buffer 2
	pKa buffer 1
	pKa buffer 2
	initial pH buffer 1
	initial pH buffer 2
	number of protons buffer 1
	number of protons buffer 2

The results do not show any significant advantage of the large feature-set (Figure 43). It took an average of 3.17 (±0.57) for the small and an average of 3.09 (±0.55) for the large feature set respectively (Table 13, 14). That means that there is no more chemical information needed, than the small feature-set provides.

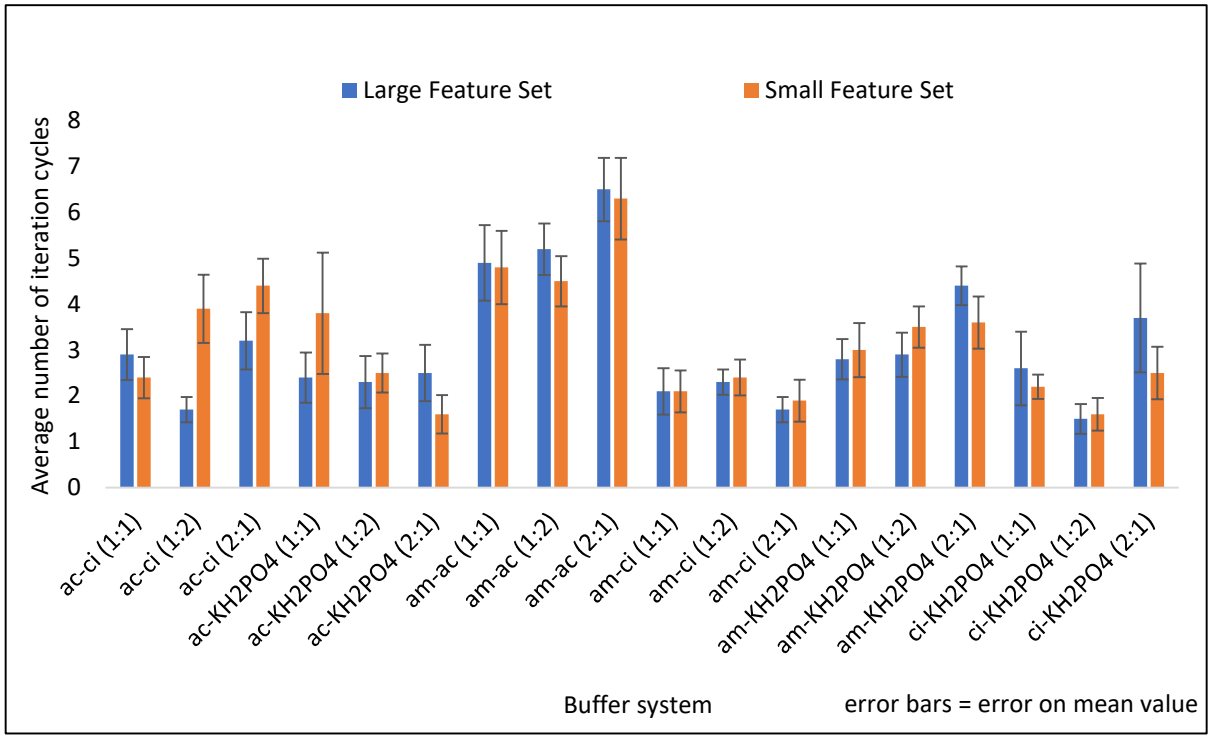

Figure 43: Comparison of average number of iteration cycles between a small and a large feature set. The small feature set only provides information about the buffer concentrations, whereas the large feature set additionally includes pKa values, initial pH values, and the number of protons of the buffers. An optimized Gaussian process (GP) was used for both feature sets. Error bars represent the error of the mean value.

Table 13: Average iteration cycles of 10 single experiments with small vs. large feature set

buffer system	average iteration cycles - small feature set	average iteration cycles - large feature set
ac-ci (1:1)	2.4	2.9
ac-ci (1:2)	3.9	1.7
ac-ci (2:1)	4.4	3.2
ac-KH_2PO_4 (1:1)	3.8	2.4
ac-KH_2PO_4 (1:2)	2.5	2.3
ac-KH_2PO_4 (2:1)	1.6	2.5
am-ac (1:1)	4.8	4.9
am-ac (1:2)	4.5	5.2
am-ac (2:1)	6.3	6.5
am-ci (1:1)	2.1	2.1
am-ci (1:2)	2.4	2.3
am-ci (2:1)	1.9	1.7
am-KH_2PO_4 (1:1)	3.0	2.8
am-KH_2PO_4 (1:2)	3.5	2.9
am-KH_2PO_4 (2:1)	3.6	4.4
ci-KH_2PO_4 (1:1)	2.2	2.6
ci-KH_2PO_4 (1:2)	1.6	1.5
ci-KH_2PO_4 (2:1)	2.5	3.7

Table 14: Result of average number of iteration cycles between a small and a large feature set. The small feature set only provides information about the buffer concentrations, whereas the large feature set additionally includes pKa values, initial pH values, and the number of protons of the buffers. An optimized Gaussian process (GP) was used for both feature sets.

feature set	average iteration cycles (± error on mean value)
small	3.2 (±0.6)
large	3.1 (±0.6)

4 Conclusion

The machine learning approach showed several advantages over conventional approaches like manual adjustment or adjustment by PID controllers. Manual adjustment has the potential of human error and is slow compared with automated systems. The results show that it is possible to adjust an unknown polyprotic buffer solution to a predefined pH value with the help of ML algorithms. Linear regression does not seem to be suitable for the present task, artificial neural network is tricky in terms of hyperparameter optimization, but performed well, and random forest showed very good results as well but with far less computational effort with reference to hyperparameter optimization. Gaussian process regression accomplished this task within three iteration cycles on average. It is not only very efficient at predicting the correct volume of acid or base to add to an unknown buffer system, but it is also very robust in terms of the lack of chemical descriptors. More chemical information only provides marginally better results in terms of iteration cycles.

The points described above offer a potential solution to the key problems when it comes to industrial scale small volume, different buffer system, high throughput pH adjustment tasks.

5 Bibliography

[1] "Arrhenius theory | Definition, Examples, & Facts | Encyclopedia Britannica." Accessed: Apr. 21, 2022. [Online]. Available: https://www.britannica.com/science/Arrhenius-theory

[2] "Bronsted-Lowry theory | Definition & Facts | Encyclopedia Britannica." Accessed: Apr. 21, 2022. [Online]. Available: https://www.britannica.com/science/Bronsted-Lowry-theory

[3] "Lewis theory | chemistry | Encyclopedia Britannica." Accessed: Apr. 21, 2022. [Online]. Available: https://www.britannica.com/science/Lewis-theory

[4] S. P. L. Sörensen, "Über die Messung und Bedeutung der Wasserstoffionen-konzentration bei biologischen Prozessen," Ergeb. Physiol. **12**, 393–532 (1912). doi: 10.1007/BF02325444.

[5] E. T. Urbansky; M. R. Schock, "Understanding, Deriving, and Computing Buffer Capacity," J. Chem. Educ., **77**, 1640 (2000). doi: 10.1021/ed077p1640.

[6] J. T. Hynes, "The protean proton in water," Nature, **397**, 565-567 (1999). doi: 10.1038/17487.

[7] K. F. Lim, "Negative pH Does Exist," J. Chem. Educ., **83**, 10, 1465 (2006). doi: 10.1021/ed083p1465.

[8] "Reference Electrodes," Chemistry LibreTexts. Accessed: Jun. 21, 2022. [Online]. Available: https://chem.libretexts.org/Bookshelves/Analytical_Chemistry/Supplemental_Modules_(Analytical_Chemistry)/Analytical_Sciences_Digital_Library/JASDL/Courseware/Analytical_Electrochemistry%3A_Potentiometry/03_Potentiometric_Theory/04_Reference_Electrodes

[9] Mettler-Toledo, "pH-Sensor für Laboranwendungen." Accessed: Jun. 21, 2022. [Online]. Available: https://www.mt.com/ch/de/home/products/Laboratory_Analytics_Browse/pH-meter/sensor/pH-sensor.html

[10] H. N. Po; N. M. Senozan, "The Henderson-Hasselbalch Equation: Its History and Limitations," J. Chem. Educ., **78**, 11, 1499 (2001). doi: 10.1021/ed078p1499.

[11] J. Berg; J. Tymoczko; L. Stryer, Stryer Biochemie, 7th ed.; Springer, 2013.

[12] "Buffers," BrainKart. Accessed: Apr. 25, 2022. [Online]. Available: https://www.brainkart.com/article/Buffers_27446

[13] "Mixtures and Buffers." Accessed: Jun. 21, 2022. [Online]. Available: https://chemed.chem.purdue.edu/genchem/topicreview/bp/ch17/mixtures.php

[14] B. J. Krieg; S. M. Taghavi; G. L. Amidon; G. E. Amidon, "In vivo predictive dissolution: transport analysis of the CO2 , bicarbonate in vivo buffer system," J. Pharm. Sci., **103**, 11, 3473–3490 (2014). doi: 10.1002/jps.24108.

[15] E. Riedel; C. Janiak, Anorganische Chemie, 8th ed.; De Gruyter, 2011.

[16] M. K. Nguyen; L. Kao; I. Kurtz, "Calculation of the equilibrium pH in a multiple-buffered aqueous solution based on partitioning of proton buffering: a new predictive formula," Am. J. Physiol. Renal Physiol., **296**, 6, 1521-1529 (2009). doi: 10.1152/ajprenal.90651.2008.

[17] "IEEE Xplore Full-Text PDF:" Accessed: Feb. 08, 2024. [Online]. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=506394

[18] I. Journal, "IRJET-Controllers used in pH Neutralization Process: A Review", Accessed: Apr. 25, 2022. [Online]. Available: https://www.academia.edu/13018117/IRJET_Controllers_used_in_pH_Neutralization_Process_A_Review

[19] M. Elarafi; S. Hisham, "Modeling and control of pH neutralization using neural network predictive controller," 2008 Int. Conf. Control Autom. Syst., 2008, doi: 10.1109/ICCAS.2008.4694329.

[20] J. A. Selekman et al., "High-Throughput Automation in Chemical Process Development," Annu. Rev. Chem. Biomol. Eng., **8**, 1, 525–547 (2017). doi: 10.1146/annurev-chembioeng-060816-101411.

[21] "Materials Acceleration Platform: Accelerating Advanced Energy Materials Discovery by Integrating High-Throughput Methods and Artificial Intelligence." Accessed: Jun. 01, 2022. [Online]. Available: https://dash.harvard.edu/handle/1/35164974

[22] M. M. Flores-Leonar et al., “Materials Acceleration Platforms: On the way to autonomous experimentation,” Curr. Opin. Green Sustain. Chem., **25**, (2020). doi: 10.1016/j.cogsc.2020.100370.
[23] B. P. MacLeod et al., “Self-driving laboratory for accelerated discovery of thin-film materials,” Sci. Adv., **6**, 20, (2020). doi: 10.1126/sciadv.aaz8867.
[24] M. Dorigo; U. Schnepf, “Genetics-based machine learning and behavior-based robotics: a new synthesis,” IEEE Trans. Syst. Man Cybern., **23**, 1, 141–154 (1993). doi: 10.1109/21.214773.
[25] A. K. Dwivedi; A. Tirkey; R. B. Ray; S. K. Rath, “Software design pattern recognition using machine learning techniques,” 2016 IEEE Region 10 Conference (TENCON), 222-227 (2016). doi: 10.1109/TENCON.2016.7847994.
[26] M. Bowling; J. Fürnkranz; T. Graepel; R. Musick, “Machine learning and games,” Mach. Learn., **63**, 211–215 (2006). doi: 10.1007/s10994-006-8919-x.
[27] J. Rzeszótko; S. H. Nguyen, “Machine Learning for Traffic Prediction,” Fundam. Informaticae, **119**, 3–4, 407–420 (2012). doi: 10.3233/FI-2012-745.
[28] F. Olsson, A literature survey of active machine learning in the context of natural language processing. Swedish Institute of Computer Science, 2009. Accessed: Jul. 30, 2022. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:ri:diva-23510
[29] “Bone cancer detection using machine learning techniques - ScienceDirect.” Accessed: Jul. 30, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128179130000171
[30] T. S. Guzella; W. M. Caminhas, “A review of machine learning approaches to Spam filtering,” Expert Syst. Appl., **36**, 7, 10206–10222 (2009). doi: 10.1016/j.eswa.2009.02.037.
[31] J. S. Shyam Mohan; H. S. Vedantham; V. C. Vanam; N. P. Challa, “Product Recommendation Systems Based on Customer Reviews Using Machine Learning Techniques,” Data Intelligence and Cognitive Informatics, 267-286 (2021). doi: 10.1007/978-981-15-8530-2_21.
[32] B. Mahesh, "Machine Learning Algorithms -A Review." 2019. doi: 10.21275/ART20203995.
[33] “What is Supervised Learning?” Accessed: Jul. 19, 2022. [Online]. Available: https://www.ibm.com/cloud/learn/supervised-learning
[34] S. Ray, “A Quick Review of Machine Learning Algorithms,” 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), 35-39 (2019). doi: 10.1109/COMITCon.2019.8862451.
[35] L. Cao; D. Russo; A. A. Lapkin, “Automated robotic platforms in design and development of formulations,” AIChE J., **67**, 5, (2021). doi: 10.1002/aic.17248.
[36] C. W. Coley; N. S. Eyke; K. F. Jensen, “Autonomous Discovery in the Chemical Sciences Part I: Progress,” Angew. Chem. Int. Ed Engl., **59**, 51, 22858–22893 (2020). doi: 10.1002/anie.201909987.
[37] J. Bai; L. Cao; S. Mosbach; J. Akroyd; A. A. Lapkin; M. Kraft, “From Platform to Knowledge Graph: Evolution of Laboratory Automation,” JACS Au, **2**, 2, 292–309 (2022). doi: 10.1021/jacsau.1c00438.
[38] X. Yan; X. Su, "Linear Regression Analysis: Theory and Computing," World Scientific (2009).
[39] L. M. Müller-Wirtz et al., “Exhaled Propofol Concentrations Correlate With Plasma and Brain Tissue Concentrations in Rats,” Anesth. Analg., **132**, 1, 110–118 (2021). doi: 10.1213/ANE.0000000000004701.
[40] C. Doucouliagos; P. Laroche, “What Do Unions Do to Productivity? A Meta-Analysis,” Ind. Relat. J. Econ. Soc., **42**, 4, 650–691 (2003). doi: 10.1111/1468-232X.00310.
[41] A. M. Legendre, "Nouvelles methodes pour la determination des orbites des cometes." Paris: F. Didot (1805).
[42] A. Kumar, “Ordinary Least Squares Method: Concepts & Examples,” Data Analytics. Accessed: Jul. 19, 2022. [Online]. Available: https://vitalflux.com/ordinary-least-squares-method-concepts-examples/
[43] I. T. Jolliffe; J. Cadima, “Principal component analysis: a review and recent developments,” Philos. Trans. R. Soc. Math. Phys. Eng. Sci., **374**, (2016). doi: 10.1098/rsta.2015.0202.

[44] K. Pearson, “On lines and planes of closest fit to systems of points in space,” The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science, **2**, 11, 559-572 (1901). doi: 10.1080/14786440109462720.
[45] C. Z. Janikow, “Fuzzy decision trees: issues and methods,” IEEE Transaction on Systems, **28**, 1, 1–14 (1998). doi: 10.1109/3477.658573.
[46] Zach, “A Simple Introduction to Random Forests,” Statology. Accessed: Feb. 12, 2024. [Online]. Available: https://www.statology.org/random-forests/
[47] S. Gupta, “Diving into the Deep learning : Random Forest Algorithm.” Accessed: Jul. 20, 2022. [Online]. Available: https://www.linkedin.com/pulse/diving-deep-learning-random-forest-algorithm-shubham-gupta
[48] E. Schulz, M. Speekenbrink, A. Krause, “A tutorial on Gaussian process regression: Modelling, exploring, and exploiting functions,” J. Math. Psychol., **85**, 1–16 (2018). doi: 10.1016/j.jmp.2018.03.001.
[49] M. P. Deisenroth, C. E. Rasmussen, J. Peters, “Gaussian process dynamic programming,” Neurocomputing, **72**, 7, 1508–1524 (2009). doi: 10.1016/j.neucom.2008.12.019.
[50] R. Sander, “Gaussian Process Regression From First Principles,” Medium. Accessed: Jul. 21, 2022. [Online]. Available: https://towardsdatascience.com/gaussian-process-regression-from-first-principles-833f4aa5f842
[51] J. C. Orduz, “An Introduction to Gaussian Process Regression” Accessed: Jul. 10, 2022. [Online]. Available: https://juanitorduz.github.io/gaussian_process_reg/
[52] J. Rice, "Mathematical Statistics and Data Analysis"; Brooks/Cole Cengage Learning, 2007.
[53] M. N. Murty and R. Raghava, “Support Vector Machines and Perceptrons: Learning, Optimization, Classification, and Application to Social Networks", M. N. Murty and R. Raghava, Springer International Publishing, 57–67 (2016). doi: 10.1007/978-3-319-41063-0_5.
[54] Y. W. Chang, C. J. Hsieh, K. W. Chang, M. Ringgaard, C. J. Lin, “Training and Testing Low-degree Polynomial Data Mappings via Linear SVM,” Journal of Machine Learning Research, **11**, 4, 1471-1491 (2010).
[55] I. Neutelings, “Neural networks.” Accessed: May 03, 2022. [Online]. Available: https://tikz.net/neural_networks/
[56] S. C. Wang, “Interdisciplinary Computing in Java Programming", The Springer International Series in Engineering and Computer Science., Springer US, 81–100 (2003). doi: 10.1007/978-1-4615-0377-4_5.
[57] J. Wu, X. Y. Chen, H. Zhang, L. D. Xiong, H. Lei, S. H. Deng, “Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimizationb,” J. Electron. Sci. Technol., **17**, 1, 26–40 (2019). doi: 10.11989/JEST.1674-862X.80904120.
[58] A. Pomberger, “First_year_report_Pomberger_final.pdf.” Apr. 28, 2021.
[59] A. Pomberger et al., “Automated pH Adjustment Driven by Robotic Workflows and Active Machine Learning,” Chem. Eng. J., **451**, (2023). doi: 10.1016/j.cej.2022.139099.
[60] K. P. Murphy, "Machine learning: a probabilistic perspective", Adaptive computation and machine learning series. Cambridge, MA: MIT Press, 2012.
[61] Genesis, “Learning Rate: Neural Network,” From The GENESIS. Accessed: Jun. 07, 2022. [Online]. Available: https://www.fromthegenesis.com/learning-rate-neural-network/
[62] “Epoch in Neural Networks", Baeldung on Computer Science.” Accessed: Jun. 07, 2022. [Online]. Available: https://www.baeldung.com/cs/epoch-neural-networks
[63] P. Sibi, S. A. Jones, P. Siddarth, “Analysis of different activation functions using back propagation neural networks,” Journal of Theoretical and Applied Information Technology, **47**, 3, 1264-1268 (2013).
[64] C. Nwankpa, W. Ijomah, A. Gachagan, S. Marshall, “Activation Functions: Comparison of trends in Practice and Research for Deep Learning,” arXiv preprint arXiv: 1811.03378 (2018)
[65] M. G. M. Abdolrasol et al., “Artificial Neural Networks Based Optimization Techniques: A Review,” Electronics, **10**, 21, (2021). doi: 10.3390/electronics10212689.

[66] M. S. B. Maind, M. P. Wankar, “Research Paper on Basic of Artificial Neural Network,” Int. J. Recent Innov. Trends Comput. Commun., **2**, 1, (2014). doi: 10.17762/ijritcc.v2i1.2920.
[67] M. Adil, R. Ullah, S. Noor, N. Gohar, “Effect of number of neurons and layers in an artificial neural network for generalized concrete mix design,” Neural Comput. Appl., **34**, 1–9 (2022). doi: 10.1007/s00521-020-05305-8.
[68] A. Krenker, J. Bešter, A. Kos, "Introduction to the Artificial Neural Networks," Artificial Neural Networks: Methodolical Advances and Biomedical Applications. InTech (2011). doi: 10.5772/15751.
[69] M. Uzair, N. Jamil, “Effects of Hidden Layers on the Efficiency of Neural networks,” IEEE 23rd Int. Multitopic Conf. INMIC (2020). doi: 10.1109/INMIC50486.2020.9318195.
[70] X. Glorot, Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 249–256 (2010). Accessed: May 04, 2022. [Online]. Available: https://proceedings.mlr.press/v9/glorot10a.html
[71] B. Ding, H. Qian, J. Zhou, “Activation functions and their characteristics in deep neural networks,” 2018 Chin. Control Decis. Conf. CCDC, (2018). doi: 10.1109/CCDC.2018.8407425.
[72] V. Nair, G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” Proceedings on the 27th international conference on machine learning (ICML-10), (2010).
[73] D. A. Clevert, T. Unterthiner, S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs),” arXiv 2015. arXiv preprint arXiv:1511.07289, 2. Accessed: May 04, 2022. [Online]. Available: https://arxiv.org/abs/1511.07289

6 Appendix

A.) Titration curves

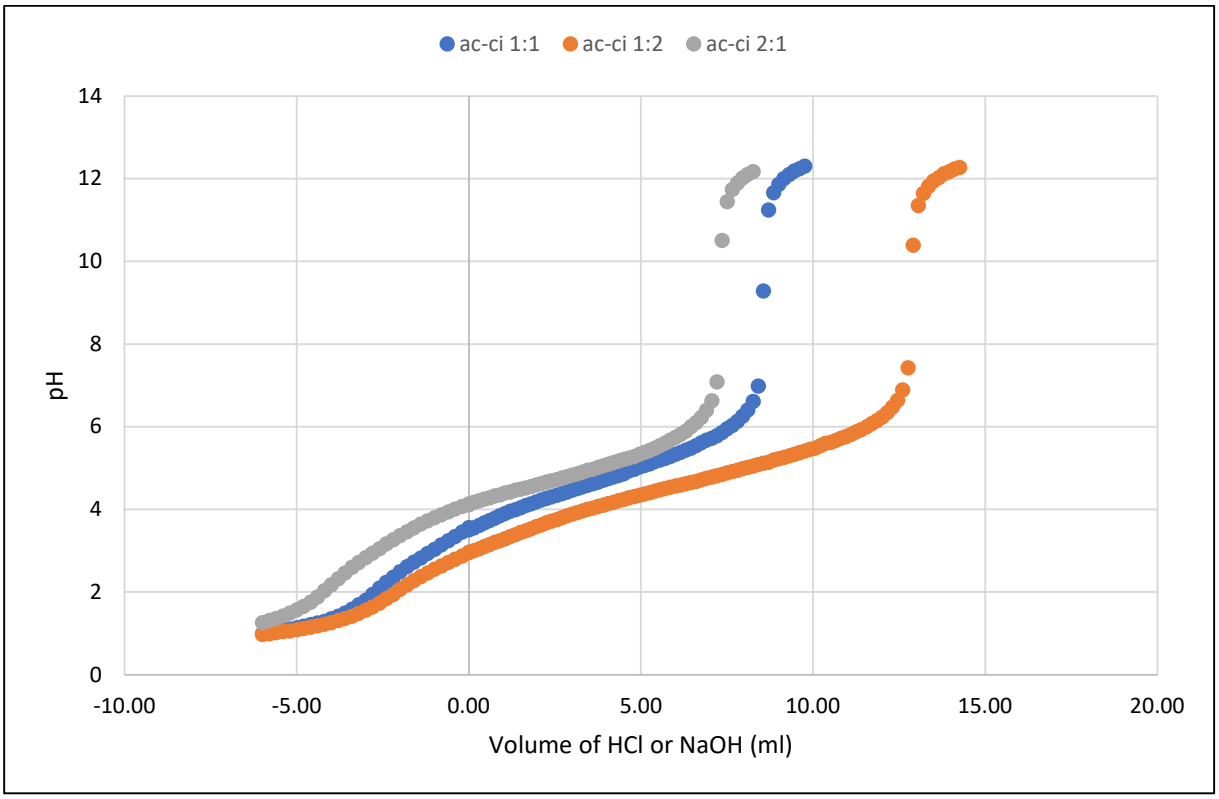

Figure 44: Titration curves for the binary buffer systems acetate-citrate with ratios of 1:1, 1:2 and 2:1. The addition of acid or base is indicated in ml on the x-axis. The addition of acid is indicated by a negative sign and the addition of base by a positive sign.

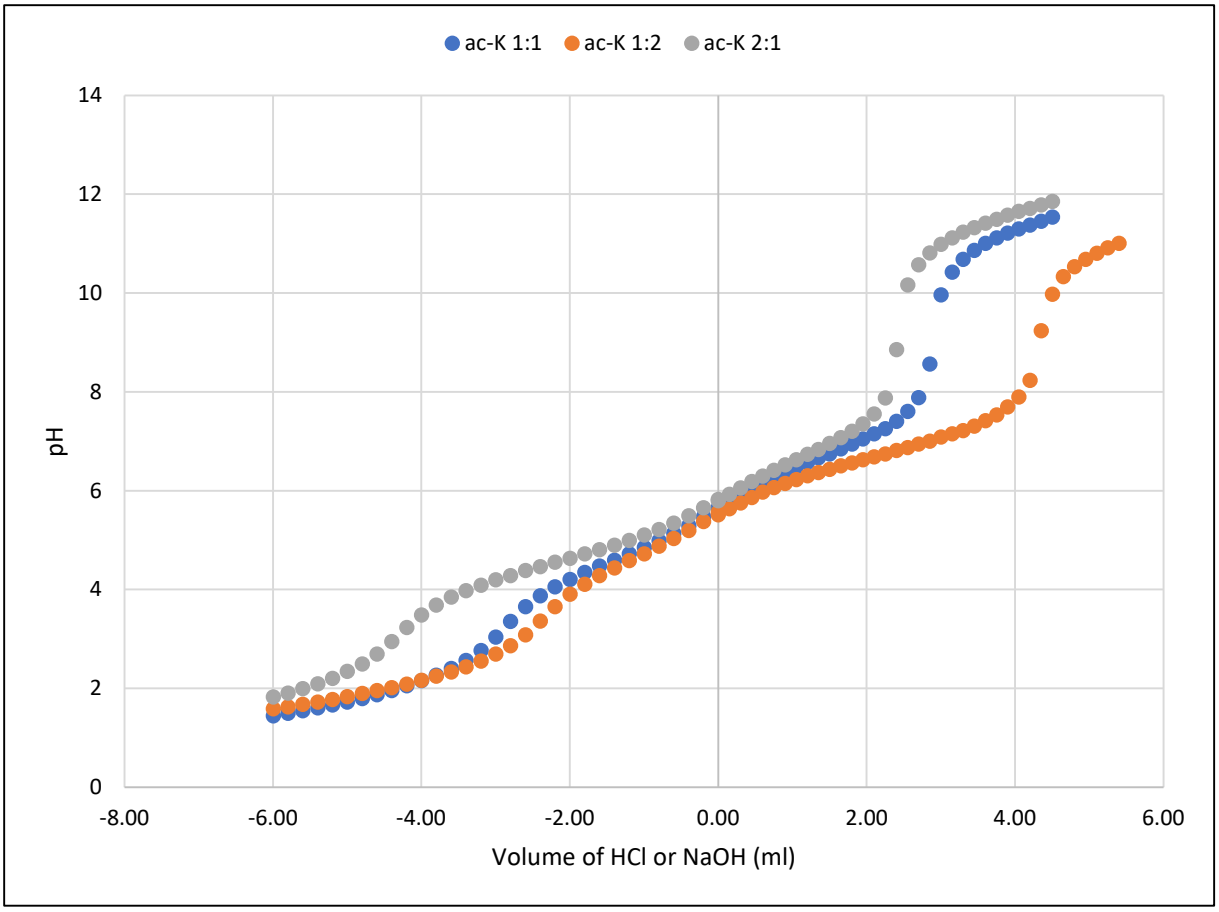

Figure 45: Titration curves for the binary buffer systems acetate-KH2PO4 with ratios of 1:1, 1:2 and 2:1. The addition of acid or base is indicated in ml on the x-axis. The addition of acid is indicated by a negative sign and the addition of base by a positive sign.

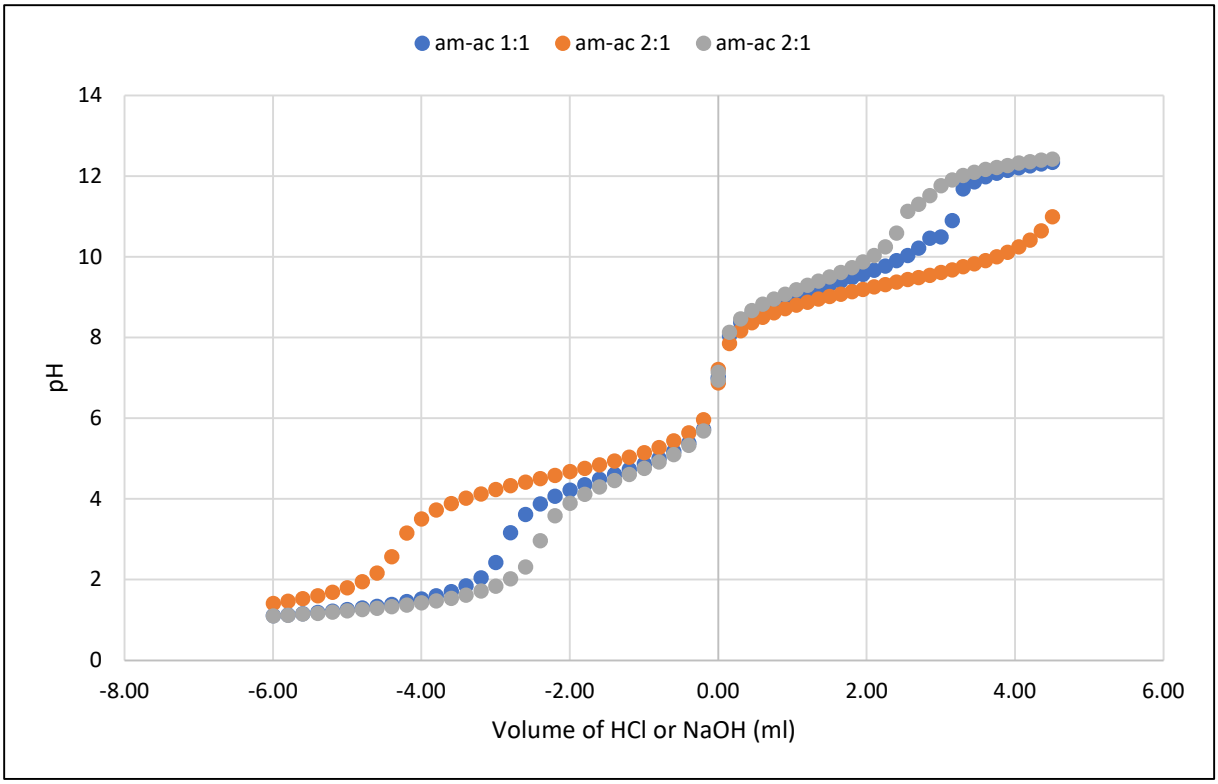

Figure 46: Titration curves for the binary buffer systems ammonium-acetate with ratios of 1:1, 1:2 and 2:1. The addition of acid or base is indicated in ml on the x-axis. The addition of acid is indicated by a negative sign and the addition of base by a positive sign.

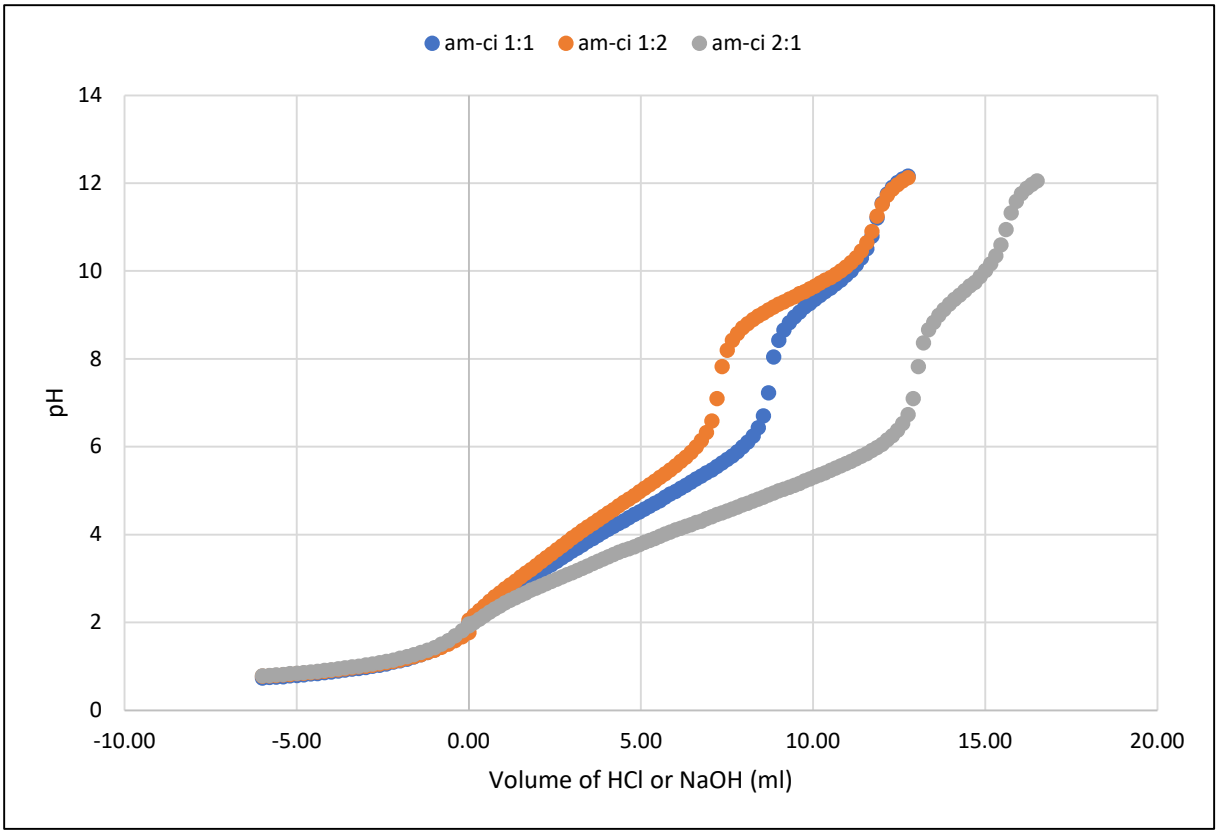

Figure 47: Titration curves for the binary buffer systems ammonium-citrate with ratios of 1:1, 1:2 and 2:1. The addition of acid or base is indicated in ml on the x-axis. The addition of acid is indicated by a negative sign and the addition of base by a positive sign.

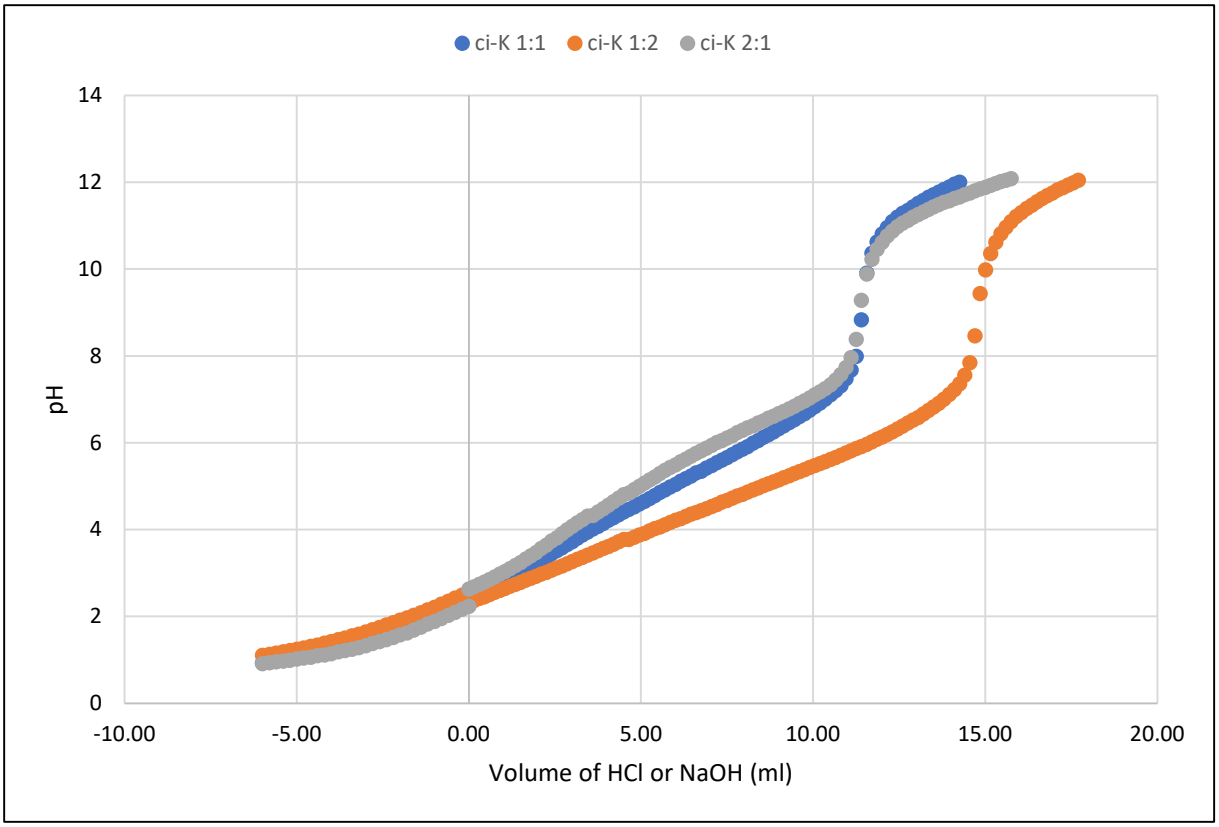

Figure 48: Titration curves for the binary buffer systems citrate-KH_2PO_4 with ratios of 1:1, 1:2 and 2:1. The addition of acid or base is indicated in ml on the x-axis. The addition of acid is indicated by a negative sign and the addition of base by a positive sign.

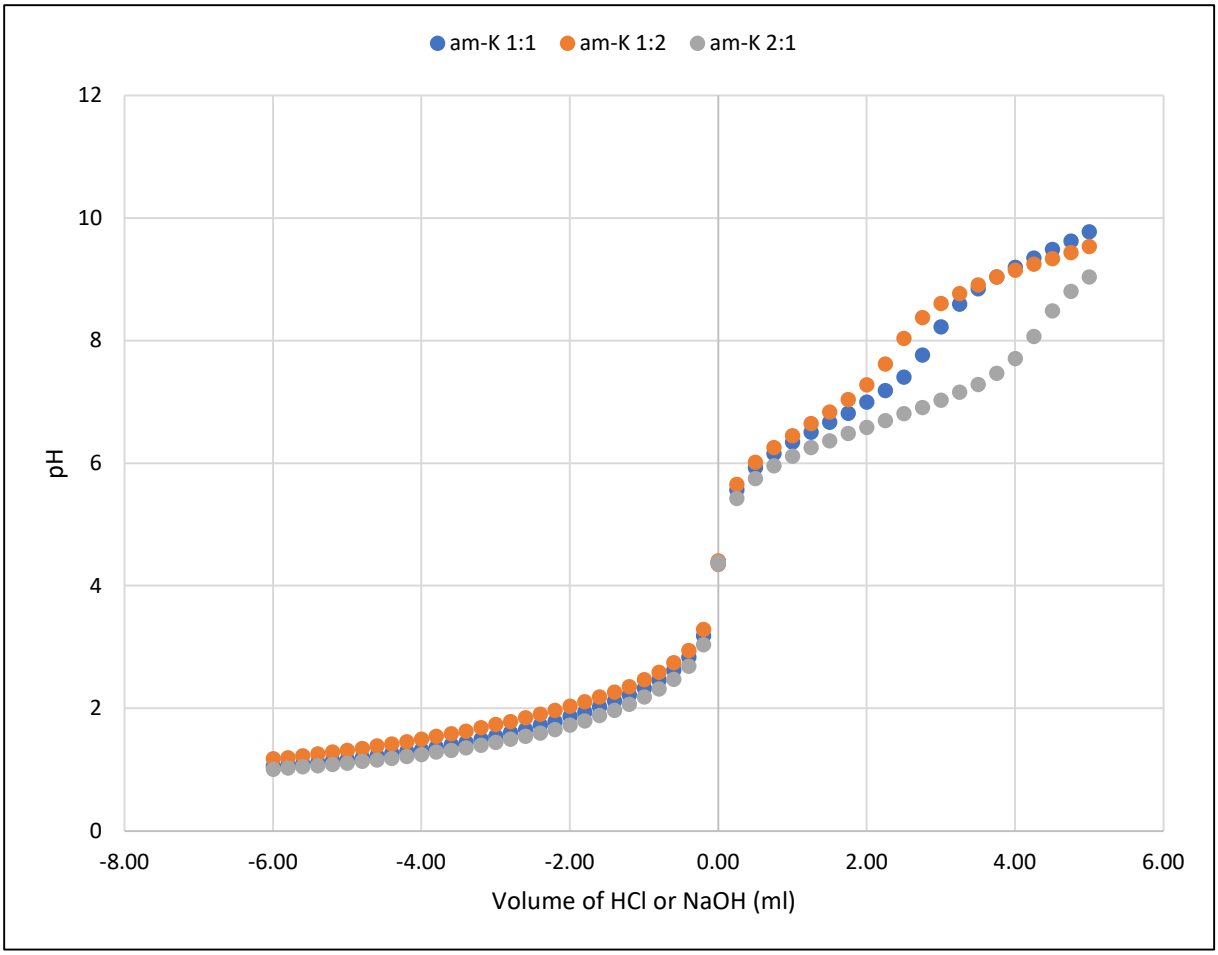

Figure 49: Titration curves for the binary buffer systems ammonium-KH_2PO_4 with ratios of 1:1, 1:2 and 2:1. The addition of acid or base is indicated in ml on the x-axis. The addition of acid is indicated by a negative sign and the addition of base by a positive sign.

B.) Activation functions

a. Sigmoid function

The sigmoid function has been successfully used in binary classification problems and modeling logistic tasks. It should be avoided for small random weights. [70] The sigmoid function is given by Equation 19. Figure 47 shows the sigmoid function.

$$f(x) = \frac{1}{1+exp^{-x}} \qquad \textit{Equation 19}$$

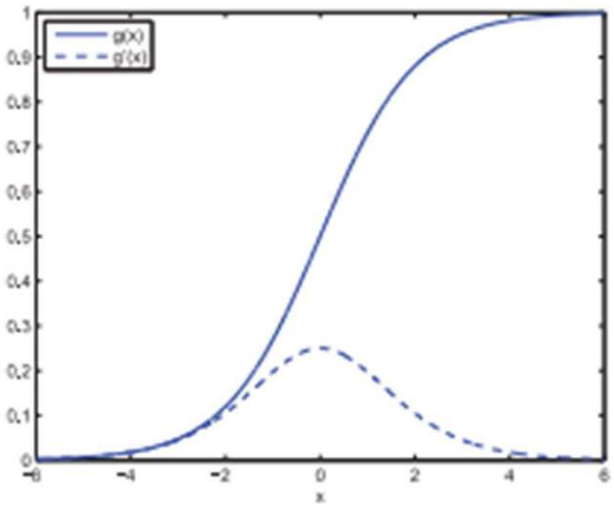

Figure 50: Sigmoid activation function [71]

b. Hyperbolic tangent function (Tanh)

The Tanh function is smoother than the sigmoid function and zero centered. Its range lies between -1 and 1 [70] and is given by Equation 20. Figure 48 shows the hyperbolic tangent function function.

$$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \qquad \textit{Equation 20}$$

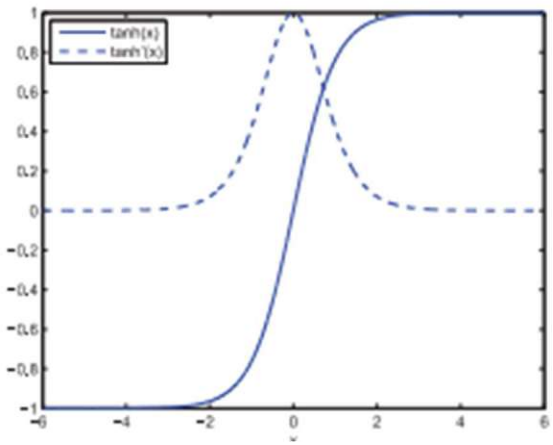

Figure 51: Tanh activation function [71]

c. Rectified linear unit function (ReLU)

The rectified linear unit activation function is the most widely used AF in artificial neural networks [72] and is given by Equation 21. Figure 49 shows the rectified linear unit function.

$$f(x) = \max(0, x) = \begin{cases} x_i, & if \ x_i \geq 0 \\ 0, & if \ x_i < 0 \end{cases} \qquad \textit{Equation 21}$$

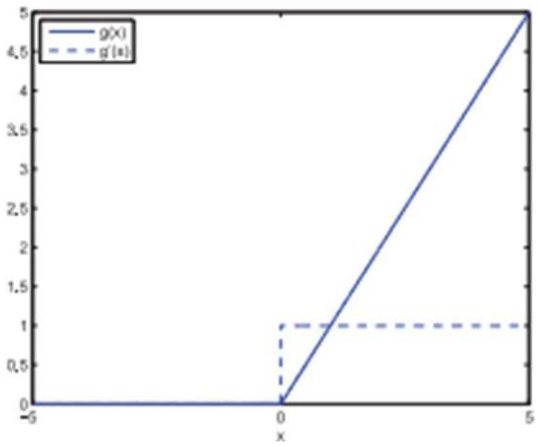

Figure 52: ReLU activation function [71]

d. Exponential linear unit (ELU)

Exponential linear units represent a good alternative to the ReLU because of reduced computational complexity which leads to improved learning speed. [73] The exponential linear unit is given by Equation 22. Figure 50 shows the exponential linear unit function.

$$f(x) = \begin{cases} x, & if\ x > 0 \\ \alpha \exp(x) - 1, & if\ x \leq 0 \end{cases} \qquad \textit{Equation 22}$$

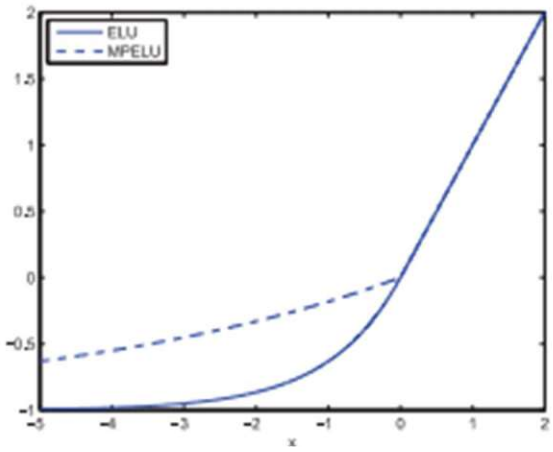

Figure 53: ELU activation function [71]

C.) Detailed benchmark results

a.) Artificial neural network

Table 15: Detailed benchmark results for the optimized artificial neural network (ANN). For each of the eighteen binary buffer systems, ten single experiments have been performed. The column "cycles" shows the number of iterative loop cycles and "pH" the final pH value after the last cycle which must reach the pre-defined pH value ±0.2.

		Experiment 1		Experiment 2		Experiment 3		Experiment 4		Experiment 5		Experiment 6		Experiment 7		Experiment 8		Experiment 9		Experiment 10	
ID	buffer system	cycles	pH	cycles	pH	cycles	pH	cycles	pH	cycles	pH	cycles	pH	cycles	pH	cycles	pH	cycles	pH	cycles	pH
1	ac-ci (1:1)	6	5.87	8	5.96	12	6.04	10	5.96	10	5.87	5	5.87	10	5.87	11	5.87	12	5.87	9	5.96
2	ac-ci (1:2)	11	5.88	7	6.01	4	6.01	12	6.01	11	6.15	7	5.88	6	5.94	6	5.82	6	5.88	10	5.94
3	ac-ci (2:1)	2	6.11	3	5.82	5	5.82	8	5.82	4	6.11	3	6.11	2	6.11	8	5.90	6	6.11	10	5.82
4	ac-KH2PO4 (1:1)	3	5.80	2	6.06	3	6.06	7	6.06	3	5.94	4	5.80	4	6.06	8	5.80	3	6.06	4	5.80
5	ac-KH2PO4 (1:2)	5	6.07	9	6.15	9	5.98	4	5.98	4	6.07	4	6.07	1	5.87	3	6.15	6	5.87	4	5.98
6	ac-KH2PO4 (2:1)	1	6.19	3	6.19	9	6.19	9	5.93	1	5.83	3	6.19	7	6.19	6	6.19	1	5.83	2	5.93
7	am-ac (1:1)	7	7.02	6	7.02	1	6.98	3	7.02	1	6.98	5	7.02	6	7.02	1	6.98	6	6.98	8	7.02
8	am-ac (1:2)	6	5.97	9	5.97	10	5.97	6	5.97	3	5.97	4	5.97	8	5.97	7	5.97	11	5.97	10	5.97
9	am-ac (2:1)	10	7.14	4	7.14	6	7.14	1	7.14	2	7.14	2	7.14	1	6.95	7	6.95	11	7.14	4	6.95
10	am-ci (1:1)	6	5.80	3	5.80	4	5.80	8	5.80	10	5.89	4	5.80	5	5.80	8	6.00	8	6.00	9	5.89
11	am-ci (1:2)	3	5.88	6	5.88	2	6.00	4	6.15	2	6.00	5	6.15	7	5.88	2	5.88	5	5.88	7	6.15
12	am-ci (2:1)	7	5.99	3	5.85	9	6.16	7	5.92	7	5.99	9	5.99	10	5.85	9	5.92	6	5.92	8	6.06
13	am-KH2PO4 (1:1)	8	6.16	4	5.93	1	5.93	4	6.16	3	5.93	6	6.16	5	5.93	8	6.16	4	6.16	7	6.16
14	am-KH2PO4 (1:2)	3	6.02	6	6.02	7	6.02	7	6.02	12	6.02	2	6.02	10	6.02	3	6.02	4	6.02	2	6.02
15	am-KH2PO4 (2:1)	4	5.96	5	6.12	2	5.96	4	5.96	8	5.96	12	5.96	6	5.96	9	5.96	10	6.12	2	6.12
16	ci-KH2PO4 (1:1)	7	6.13	3	6.13	4	5.92	2	5.86	1	5.80	6	6.06	8	6.06	5	5.86	6	6.06	3	5.86
17	ci-KH2PO4 (1:2)	4	5.97	5	5.97	4	5.97	10	5.97	3	5.87	1	5.92	8	5.82	1	5.92	5	5.87	5	5.97
18	ci-KH2PO4 (2:1)	5	6.12	6	6.17	4	6.17	2	5.88	10	6.17	4	5.82	5	6.01	4	6.06	7	5.95	5	6.12

b.) *Random forest*

Table 16: Detailed benchmark results for the optimized random forest (RF). For each of the eighteen binary buffer systems, ten single experiments have been performed. The column "cycles" shows the number of iterative loop cycles and "pH" the final pH value after the last cycle which must reach the pre-defined pH value ±0.2.

		Experiment 1		Experiment 2		Experiment 3		Experiment 4		Experiment 5		Experiment 6		Experiment 7		Experiment 8		Experiment 9		Experiment 10	
ID	buffer system	cycles	pH	cycles	pH	cycles	pH	cycles	pH	cycles	pH	cycles	pH	cycles	pH	cycles	pH	cycles	pH	cycles	pH
1	ac-ci (1:1)	1	5.87	4	5.87	4	5.87	8	5.87	3	5.96	3	5.87	4	5.96	4	5.87	4	5.96	5	5.96
2	ac-ci (1:2)	1	6.01	7	5.88	3	6.01	6	6.01	2	5.94	4	6.01	3	5.94	3	6.08	2	6.08	4	5.94
3	ac-ci (2:1)	7	5.90	1	6.01	7	6.11	5	6.01	5	5.82	4	5.82	6	5.82	4	5.90	5	6.01	3	5.90
4	ac-KH2PO4 (1:1)	1	6.18	8	6.18	3	6.18	3	5.94	1	5.80	1	6.18	2	5.94	2	5.94	2	5.94	3	6.18
5	ac-KH2PO4 (1:2)	5	5.98	4	5.98	3	6.07	2	6.15	1	5.87	1	6.15	8	6.07	6	6.15	3	6.15	1	5.98
6	ac-KH2PO4 (2:1)	7	5.80	6	5.83	1	5.80	3	6.19	3	6.06	1	6.06	4	5.83	13	6.19	5	6.06	1	6.19
7	am-ac (1:1)	2	7.02	4	6.98	6	6.98	2	7.02	3	6.98	4	6.98	4	6.98	3	6.98	5	6.98	2	6.98
8	am-ac (1:2)	4	5.97	7	5.97	4	5.97	6	5.97	8	5.97	3	5.97	6	5.97	4	5.97	4	5.97	8	5.97
9	am-ac (2:1)	4	6.95	3	7.14	4	7.14	3	7.14	4	7.14	1	6.95	5	6.95	3	6.95	6	7.14	2	6.95
10	am-ci (1:1)	3	5.89	5	6.11	4	5.80	2	5.89	1	5.89	1	5.89	1	6.00	3	5.89	1	5.89	12	6.11
11	am-ci (1:2)	5	6.15	4	5.88	5	5.88	4	6.00	5	6.15	3	6.00	6	5.88	3	6.00	5	5.88	3	5.88
12	am-ci (2:1)	5	5.85	4	5.85	4	5.99	4	5.92	5	6.06	4	5.92	5	5.92	5	5.92	4	6.16	5	5.92
13	am-KH2PO4 (1:1)	4	5.93	4	6.16	1	6.16	3	5.93	5	6.16	2	6.16	4	6.16	5	5.93	5	6.16	5	6.16
14	am-KH2PO4 1:2	6	6.02	4	6.02	2	6.02	4	6.02	4	6.02	9	6.02	4	6.02	5	6.02	4	6.02	3	6.02
15	am-KH2PO4 (2:1)	8	5.96	5	5.96	4	6.12	3	6.12	3	6.12	2	5.96	4	6.12	5	5.96	4	5.96	2	5.96
16	ci-KH2PO4 (1:1)	3	6.00	4	6.06	4	5.92	6	6.00	3	6.06	1	5.80	4	5.86	2	6.00	3	6.00	1	6.06
17	ci-KH2PO4 (1:2)	1	6.03	3	5.87	2	5.82	1	6.09	4	5.97	5	5.97	4	6.03	3	5.92	5	6.03	4	6.03
18	ci-KH2PO4 (2:1)	4	6.01	3	6.12	3	6.12	3	5.88	5	6.12	3	5.88	9	6.06	1	6.06	4	6.01	1	6.17

c.) *Gaussian process*

Table 17: Detailed benchmark results for the optimized gaussian process (GP). For each of the eighteen binary buffer systems, ten single experiments have been performed. The column "cycles" shows the number of iterative loop cycles and "pH" the final pH value after the last cycle which must reach the pre-defined pH value ±0.2.

		Experiment 1		Experiment 2		Experiment 3		Experiment 4		Experiment 5		Experiment 6		Experiment 7		Experiment 8		Experiment 9		Experiment 10	
ID	buffer system	cycles	pH	cycles	pH	cycles	pH	cycles	pH	cycles	pH	cycles	pH	cycles	pH	cycles	pH	cycles	pH	cycles	pH
1	ac-ci (1:1)	5	5.96	1	6.14	3	5.87	2	5.96	1	5.96	2	5.96	2	5.87	6	5.96	5	5.96	5	5.96
2	ac-ci (1:2)	7	5.88	2	5.88	8	5.94	4	5.82	5	5.88	6	5.82	1	6.01	6	5.88	5	5.82	1	5.82
3	ac-ci (2:1)	7	5.90	3	5.90	2	6.01	1	5.90	6	6.01	4	5.82	5	5.82	2	6.11	6	5.90	4	5.90
4	ac-KH2PO4 (1:1)	2	5.80	2	5.80	3	5.94	4	5.80	6	6.06	3	5.94	1	5.94	1	5.94	1	5.94	1	6.06
5	ac-KH2PO4 (1:2)	2	6.15	1	5.98	3	6.07	1	6.07	1	6.15	1	5.98	6	5.87	4	5.87	1	6.07	3	5.87
6	ac-KH2PO4 (2:1)	2	5.80	4	5.93	1	6.06	1	5.93	1	6.06	2	5.80	7	5.93	2	5.93	2	5.93	3	5.93
7	am-ac (1:1)	3	7.02	2	7.02	4	6.98	9	6.98	1	7.02	6	6.98	5	6.98	5	6.98	7	6.98	7	7.02
8	am-ac (1:2)	7	5.97	6	5.97	7	5.97	6	5.97	5	5.97	4	5.97	3	5.97	6	5.97	6	5.97	2	5.97
9	am-ac (2:1)	2	7.14	3	6.95	2	6.95	2	7.14	6	6.95	2	7.14	7	6.95	7	6.95	2	6.95	7	6.95
10	am-ci (1:1)	3	6.11	4	5.89	4	6.00	3	5.89	5	5.80	4	5.89	2	5.80	4	5.80	2	6.00	3	5.89
11	am-ci (1:2)	3	6.00	7	5.88	1	6.15	3	5.88	1	5.88	3	5.88	3	6.00	4	6.00	2	5.88	4	6.00
12	am-ci (2:1)	5	5.92	2	5.99	2	6.16	4	5.92	5	5.92	6	5.92	4	6.06	3	5.85	2	5.99	5	5.99
13	am-KH2PO4 (1:1)	1	6.16	4	5.93	1	6.16	3	6.16	4	5.93	1	5.93	2	6.16	2	5.93	4	5.93	1	6.16
14	am-KH2PO4 (1:2)	1	6.02	4	6.02	1	6.02	6	6.02	4	6.02	2	6.02	4	6.02	1	6.02	5	6.02	3	6.02
15	am-KH2PO4 (2:1)	3	5.96	3	6.12	5	5.96	4	6.12	4	5.96	5	5.96	3	5.96	4	5.96	5	5.96	4	6.12
16	ci-KH2PO4 (1:1)	1	6.19	4	5.92	2	6.00	1	5.92	2	6.13	1	6.19	3	5.86	2	6.00	1	5.80	2	5.92
17	ci-KH2PO4 (1:2)	2	5.97	3	5.87	4	5.97	2	5.92	2	5.97	2	5.92	1	6.03	2	5.97	1	5.97	5	5.97
18	ci-KH2PO4 (2:1)	4	5.95	1	6.01	2	6.01	3	5.88	2	5.82	1	6.01	1	6.01	1	5.95	3	5.95	1	5.88