Rowan University
Department of Chemical Engineering
201 Mullica Hill Road
Glassboro, NJ, 08028
October 31, 2023

Mr. Jeffrey Perez
American Institute of Chemical Engineers
120 Wall St, 23rd Floor
New York, NY, 10005

Dear Mr. Perez,

Attached please find the Rowan UEF SEC grant awardee team's final report, which outlines the work done over the summer and fall of 2023 and the previous work that it continues. This project's work builds upon the foundations established by prior studies to create working models and algorithms to predict chemicals' life cycle inventories (LCIs). Data collection was done to create a model using an eXtreme Gradient Boosting (XGBoost) and Advanced Neural Networks (ANN) Machine Learning algorithm. The results of these models are analyzed, tested, and applied to a case study from literature. Conclusions regarding this work were drawn with thoughts on possible future work and development. If you have any questions or require any clarification with the report, please feel free to contact the team at the emails below.

Sincerely,

Ethan Shumaker
shumak96@students.rowan.edu

Jared Longo
Longoj89@students.rowan.edu

John Pazik
pazikj95@students.rowan.edu

Matt Conway
conway64@students.rowan.edu

Ethan Blanda
blanda26@students.rowan.edu

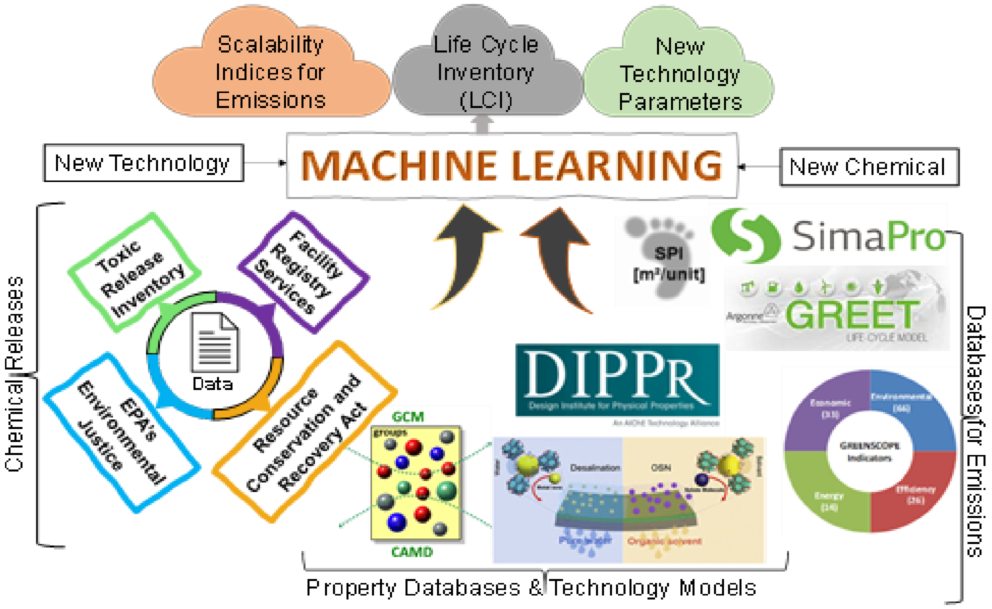
Machine Learning for Sustainable Chemicals and Processes

Abstract

Sustainability in chemical processes is paramount to increasing sustainability across society. It is therefore important to include sustainability assessments throughout the design process, especially in the early stages. This is difficult, though, due to the lack of specificity and information at that point, which makes the evaluation of different sustainability metrics difficult. Improvements to sustainability become more difficult as the design progresses, however, as earlier decisions limit future research and considerations (Argoti et al.). Better analysis in these early stages is therefore necessary to overall sustainability with consideration for a multitude of factors that impact the processes and their chemicals' environmental effects (Finnveden et al.). Machine Learning (ML) may be a critical tool in early-stage process design, as it has been shown to accurately predict the effects of novel chemicals despite little available information (Karka et al.).

This work aims to develop a ML algorithm that accurately forecasts the effects of chemicals in the environment, especially in relation to human health, ecosystem quality, climate change, and resource utilization. It also introduces an innovative approach to cradle-to-cradle life cycle assessments wherein the solvent is reclaimed for future use in the disposal phase. The study employed XGBoost and artificial neural networks (ANN). Data was sourced from various chemical databases, including Ecoinvent, and the feature set included over 200 molecular descriptors and 23 thermodynamic properties for each element. A stepwise feature selection process was used to reduce the number of features from 223 to 10. Following hyperparameter tuning, the model's performance was assessed on the test set using the R-squared and Root-Mean-Squared-Error (RMSE). To illustrate the model's application, a case study was carried out on a cradle-to-cradle LCA of acetone. The gate-to-gate phase is modeled through the valorization of red wine pomace through solvent extraction, considering emissions during the use phase, including energy-related emissions and fugitive emissions from separation processes. Finally, the end-of-life phase of the acetone consisted of the recovery of the solvent through distillation for reuse. Any requests for access to the code files and documentation can be directed to the authors.

Keywords: Machine Learning, Environmental Impact, Life Cycle Inventories, Sustainability



Graphical Abstract demonstrating the overall scope of the project

Contents

1. Introduction

As of 2017, the chemical industry was the world's second-largest manufacturing industry (UN Environment Programme, 2019). Between the years 2017 and 2030, the global chemical market is projected to double. The projected growth brings about concerns for the environment due to the associated growth in chemical waste with the growth in the market (US EPA, 2013). As with any manufacturing industry, there is waste associated with production due to inefficiencies in the process. In the case of the chemical industry, these inefficiencies can be attributed to inefficient mixing, the quality of the raw materials, and inappropriate process equipment being utilized, among others. A common way to quantify the amount of waste produced is using an E-factor (A. Sheldon, 2007), as given by Equation 1. Table 1 presents a summary of different E-factors in the chemical industries. As shown in Table 1, the pharmaceutical industry is the main source of solvent waste due to the multiple purification steps associated with producing high purity active pharmaceutical ingredients (API).

$$\text{E-factor} = \frac{Total\ mass\ waste\ produced}{Total\ mass\ product\ produced}\ (1)$$

Table 1: E-factors in various industries (A. Sheldon 2007)

Industry	Tons of Product/Year	E-factor
Oil Refining	10^6-10^8	<0.1
Bulk Chemicals	10^4-10^6	<1-5
Fine Chemicals	10^2-10^4	5-50
Pharmaceuticals	10-10^3	25-100

Furthermore, the growing chemical market is an indication of the development of novel chemicals and processes. New chemicals often lack the data needed to accurately quantify their potential environmental impact, requiring experimentation or computationally intensive simulations to acquire them. These expensive and time-consuming methods can discourage many small-scale production facilities from being able to consider greener or safer alternatives.

Additional impacts to the environment involve the wasteful use of solvents leading to the buildup of hazardous waste. With many solvents being treated as single use consumables, waste accumulation is bound to occur. To combat this, solvent recovery techniques can be utilized to purify and recirculate used solvent back into a process. Proper recovery techniques require proper analysis of the raw waste stream along with either purity or recovery specifications for the refined outlet stream. The selection of recovery technologies to achieve this goal can be performed through optimization software to minimize cost and environmental impact of operation (Aboagye et al., 2021). To quantify environmental impact a life cycle analysis (LCA) needs to be performed on the raw materials and the resulting products.

Life cycle analysis (LCA) is a systematic analysis of potential environmental impacts of raw materials, products, or services during their entire life cycle (Karka et al., 2019). Sustainability assessment (SA) is a complex evaluation methodology that involves integrated perspectives from environmental, economic, public health, social, and demographics that extend beyond the traditionally used pure technoeconomic analysis (TEA) (Sala et al., 2015). Both LCA and SA are data-intensive processes that can benefit from advanced statistics and machine learning (ML) methods. LCA and SA require life cycle inventory (LCI) data for all chemicals and technologies involved in a process, which is not readily available for new alternative chemicals, solvents, and technologies. Furthermore, scalability correlations for costs and emission factors, and operating parameters are essential to analyze alternative chemicals processed in novel technologies to compare them to traditional practices during the initial process synthesis and design stage (Mercado & Cabezas, 2016).

ML has before been shown to help improve energy efficiency and predict carbon footprints of corporations (Narciso & Martins, 2020, and Nguyen et al., 2021). This work, though, aims to prove ML helpful in predicting accurate LCIA values for chemicals, especially for new molecules and process technologies. This innovative method can subsequently lead to safer, more sustainable alternative chemicals and circular process design. Predicting LCIA metrics and scale-up factors for environmental indicators at the preliminary stages of process design can be crucial for advancing green and sustainable processes for novel chemicals and processes. Specifically, using ML would allow for this data to be acquired without the time and cost commitment required currently. The proposed solution is to use data-driven approaches to predict the LCIA metrics of chemicals and to develop strategies to predict scaling factors. In this report, the first solution is reached by using machine learning approaches to predict LCIA metrics, specifically, endpoint impacts of solvents based on their thermodynamic and molecular properties. These properties have been used before to predict these impacts using mixed-integer programming, but this work aims to achieve better results using ML (Calvo-Serrano et al. 2017).

This report is organized into four main sections. Section one is the introduction which discusses the motivation for this research. In section two, the methodology used in this research is described. Section three presents and discusses the results from the research. Finally, some conclusions and future works are presented in the last section.

2. Methods: Developing ML Algorithms for LCIA Metric Prediction

This section first discusses the data collection step and corresponding data preprocessing techniques used. Next, the two ML algorithms used in this research are described, ending with a discussion about the steps involved in the tuning of the hyperparameters for each model. Figure 1 presents the overall workflow, which is organized into two general steps: data preprocessing and model building.

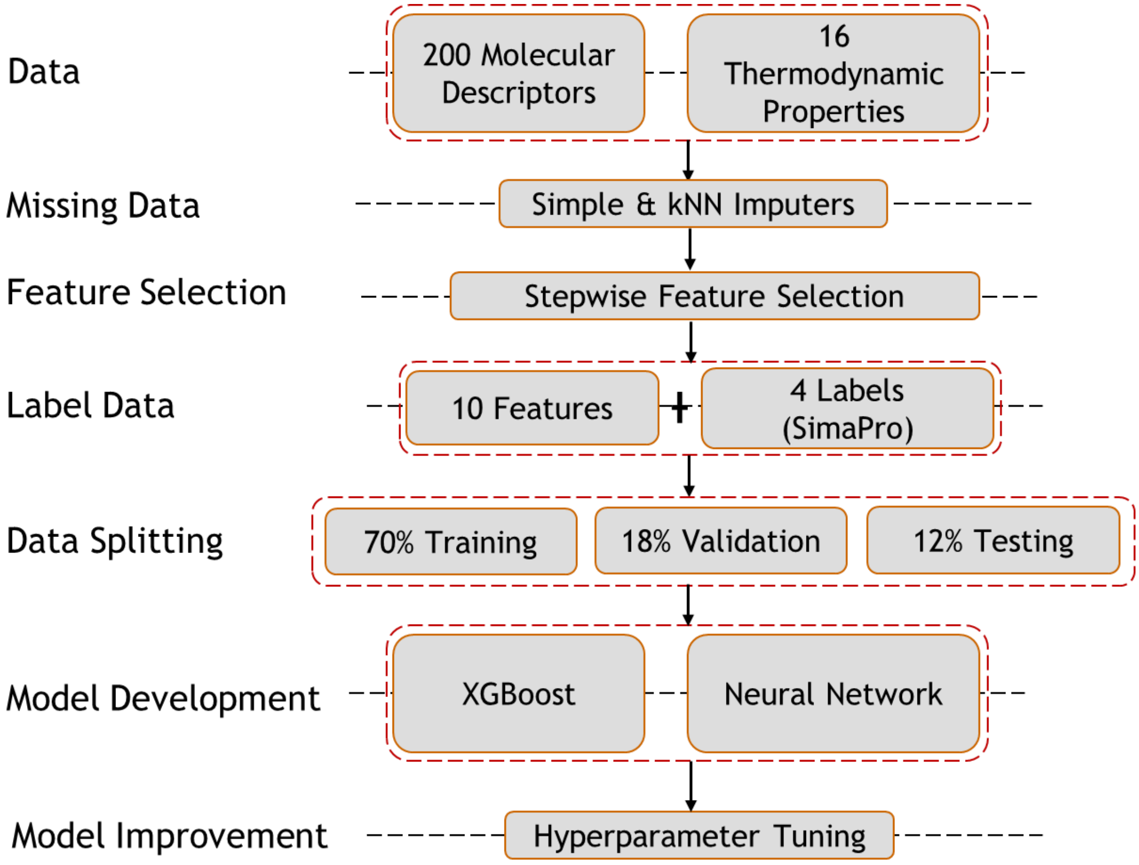


Figure 1. Overview of ML model workflow

2.1 Data Collection

To build any machine learning model, the first step is data acquisition. Developing a fast and reliable method for acquiring the necessary data is integral for model development because a vast amount of data is needed to train and test a ML model; Models trained with larger data sets make better predictions. To acquire this data with a reasonable amount of effort and time, the model uses an automated data collection process. The data acquisition step was divided into two main parts: the input feature dataset and the output label dataset. The features dataset is the independent input set from which a prediction is made. The label dataset is what the ML model is trying to predict. In this work, the feature set is comprised of both thermodynamic and molecular descriptor properties of the chemicals, while the label dataset consists of four endpoint impact assessment metrics: human health impact (HHI), ecosystem quality impact (EQI), climate change impact (CCI), and resource utilization impact (RUI).

Organic molecules are a majority of the chemicals in the CAS database; because of their prevalence in the chemical industry, a training set focused on organic molecules was used. In total, sixteen thermodynamic properties and two hundred molecular descriptors were collected, as well as each chemical's name, formula, CAS number, and SMILES structure. Priority was placed on properties that directly affected the energy costs of certain processes, such as heat of vaporization, enthalpies of formation, and boiling points. The three main python libraries used in feature extraction of the thermodynamic properties were the “chemicals”, “thermo”, and “pubchempy” packages. Each package connects to databases such as PubChem, National Institute of Standards and Technology (NIST), and other reputable and publicly available databases. To collect the molecular descriptors for our training set, the python package "rdkit" was used. RDKit is an open-source toolkit developed primarily for cheminformatics. Its core algorithms are in C++ but there are python wrappers such as packages developed that can be used to access the molecular descriptor of chemical provided the SMILEs code for the specific chemical is known. A total of 200 molecular descriptors can be collected from rdkit, which when combined with the 23 thermodynamic features described earlier yields 223 features collected for 502 organic chemicals.

The next step was to collect the corresponding life cycle impact assessment metrics for all the 350 chemicals. As research progressed, multiple methods were used. SimaPro, a software which uses the Ecoinvent database, was used in the initial stages of this data collection task. SimaPro is a well-known software used for performing life cycle assessment of products and processes. In SimaPro, this project used Ecoinvent, cut-off by unit, as the base database. Once the production process of the chemical is obtained, the IMPACT 2002+ method of impact assessment was used to quantify the emissions, midpoint impacts, and the endpoint impact for the production process. In all analysis, a functional unit of 1 kg of the organic compound was used. Thus, from SimaPro the HHI, EQI, CCI, and RUI metrics for each chemical were acquired, giving a combined feature and label set of 220 (216 features, 4 labels) with 350 data points. As data collection continued, direct access to the Ecoinvent database was used to add an additional 152 data points containing the same set of features to the training set, for a total of 502 data points. Table 2 is an example of the type of data extracted for the research endeavor.

Table 2: Example of data collected for each chemical

Chemical Name	Chemical Formula	CAS #	SMILES	Heat of Vaporization (kJ/kg)	Heat Capacity (kJ/kgK)	Number of Hetero-atoms	Climate Change (kg CO2-eq)
1-BROMOPROPANE	C3H7BR	106-94-5	CCCBr	32200	1.09	1	4.677
1-CHLORO-1,1-DIFLUOROETHANE	C2H3ClF2	75-68-3	CC(F)(F)Cl	20400	1.31	3	3.979
1,1,1-TRICHLOROETHANE	C2H3Cl3	71-55-6	CC(Cl)(Cl)Cl	29900	1.08	3	2.085
1,1,2,2-TETRACHLOROETHANE	C2H2CL4	79-34-5	ClC(Cl)C(Cl)Cl	44400	0.99	4	2.894
1,2-DIBROMOETHANE	C2H4BR2	106-93-4	BrCCBr	40030	0.72	2	0.934
1,2-DICHLOROETHANE	C2H4CL2	107-06-02	ClCCCl	35100	1.31	2	2.813
1,2,4-TRIMETHYLBENZENE	C9H12	95-63-6	Cc1ccc(C)c(C)c1	47800	1.79	0	2.139
1,3-BUTADIENE	C4H6	106-99-0	C=CC=C	20800	2.29	0	1.090
1,3-PHENYLENEDIAMINE	C6H8N2	108-45-2	Nc1cccc(N)c1	65400	2.56	2	18.584
1,4-DICHLOROBENZENE	C6H4Cl2	106-46-7	Clc1ccc(Cl)cc1	46900	1.20	2	2.353

2.2 Clustering

Once the data was obtained, further work needed to be done with it for the data to be optimal for the final machine learning model. Much of the data was missing and needed to be filled in since the model cannot use null values. A clustering algorithm was determined to be the best way to fill in the missing data, and the following flowchart describes the general process for creating a new dataset without any missing values.

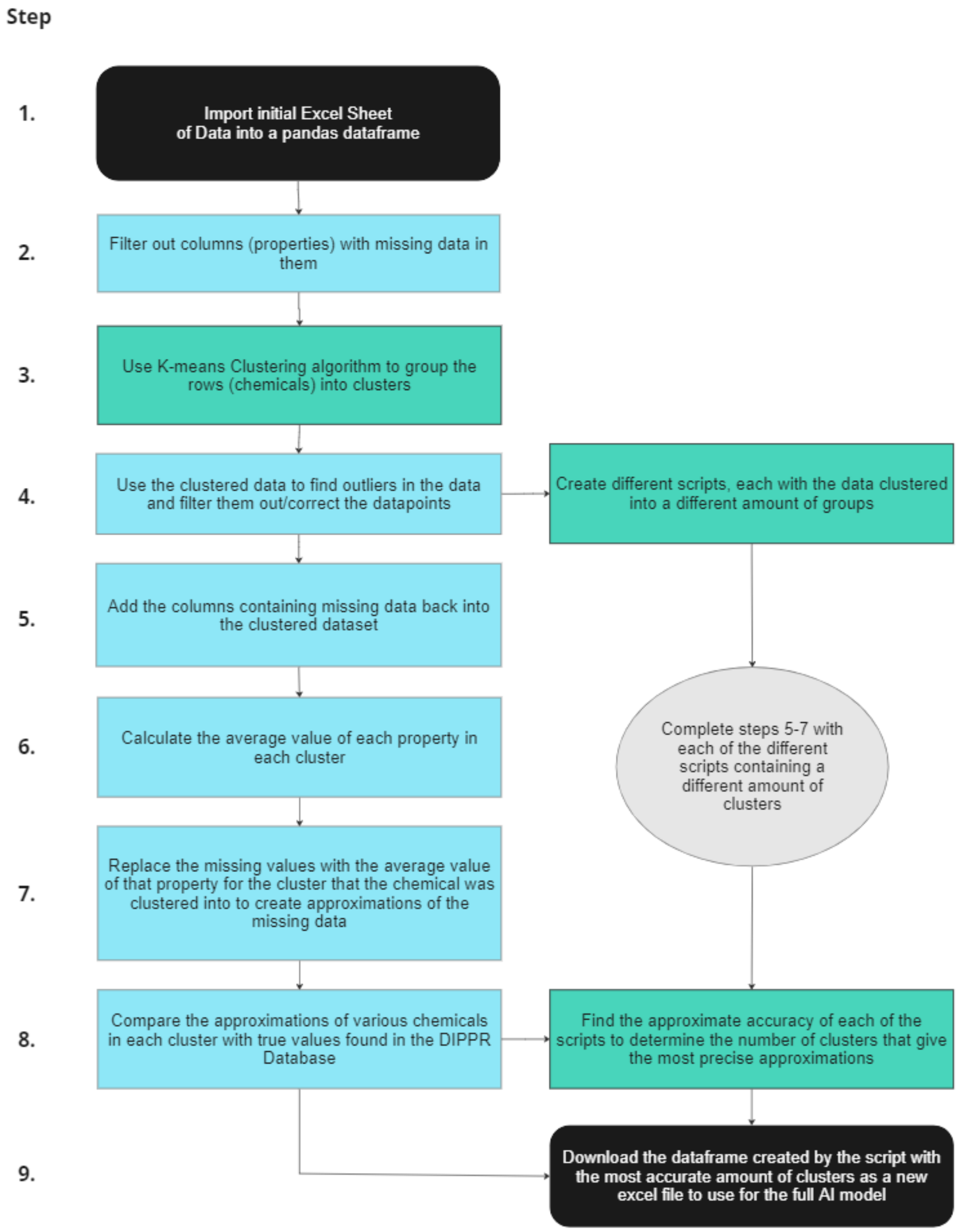

Step
1.
Import initial Excel Sheet of Data into a pandas dataframe
2.
Filter out columns (properties) with missing data in them
3.
Use K-means Clustering algorithm to group the rows (chemicals) into clusters
4.
Use the clustered data to find outliers in the data and filter them out/correct the datapoints
Create different scripts, each with the data clustered into a different amount of groups
5.
Add the columns containing missing data back into the clustered dataset
6.
Calculate the average value of each property in each cluster
Complete steps 5-7 with each of the different scripts containing a different amount of clusters
7.
Replace the missing values with the average value of that property for the cluster that the chemical was clustered into to create approximations of the missing data
8.
Compare the approximations of various chemicals in each cluster with true values found in the DIPPR Database
Find the approximate accuracy of each of the scripts to determine the number of clusters that give the most precise approximations
9.
Download the dataframe created by the script with the most accurate amount of clusters as a new excel file to use for the full AI model

Figure 2. Flowchart describing general K-Means Clustering Process

The clustering algorithm used was K-means clustering. In the dataset, rows represent chemicals and columns represent the various thermodynamic properties and molecular descriptor values. Columns with missing data were temporarily removed to create a smaller dataset containing all the chemicals and a subset of the properties (where there were no missing data values). This subset was then fed into the algorithm to create different groupings of chemicals that the algorithm determined were similar. Once clustered, some chemicals appeared as outliers and were alone in a cluster. These chemicals were manually reviewed for error in the data scraping process. The algorithm was then rerun with the new fixed dataset.

After looking at the graph of the accuracy of the clusters compared to the number of them, the elbow method was used to determine that the optimal number of clusters was between four and ten clusters. Nearly identical scripts were then made for each of the different clustered datasets. For example, when looking at the script where the chemicals were clustered into four groups, the algorithm sorted each of the five hundred chemicals into one of four groups based on the properties that they all had values for.

After the now clustered subset of data is created, the columns with missing data are added back to the subset. The dataset is then split up by cluster (meaning if the data was clustered into four groups, then the dataset will be separated into four separate datasets).
Within each separate dataset, the average for each column is then taken and replaces the missing data in that column. Every column in every separate data set undergoes this process, removing all empty data by replacing it with the column average in that cluster. Once this is complete, the separate datasets then recombine to create a new dataset with no missing values. The average values act as estimates for the missing data.

To check the clustered values' accuracy, the estimated density for fifteen different chemicals were chosen and compared to the known density values found in the DIPPR database. Density was chosen because it was a property where about half of the chemicals needed for it to be estimated and the known densities are readily available to cross compare accuracy with. Each script with a different number of clusters undergoes this process to determine the average percent error, and the one with the lowest percent error is determined as the most accurate and to be the data used moving forward. That final version is then ported back into an Excel file to be used by the ANN model.

As for the results of the clustered data, six clusters are optimal for the data obtained. When clustering the data into more clusters than six, there was too much missing data and some of the columns within clusters had no values to use for estimations. Using six clusters, the density values cross-referenced had an average percent error of 16.74%. While this is not ideal, it is a much better estimate than estimating the missing data without clustering it.

2.3 Developing the ML Algorithm

When training a machine learning model, the data must be split into a testing and a training set with also optionally a cross validation set. The training data is used to fit the model; this data is put into the program to create and adjust the variables used for the model to generate predictions. The cross-validation set is then used for the program to compare the model fitted with the training data set. The program then runs the model with the cross validation set and adjusts the weights according to how accurate the predicted outputs are to improve the model. Finally, the testing set is used to run the model and compare the predicted outputs to the testing set's known outputs to find the accuracy of the model.

The ANN model for machine learning was used. It can be used for a wide variety of applications and is easily accessible for all Python users. The programming language chosen was Python because of its ease of usage, wide support community, and ability to interface with different programming and web service platforms.

2.3.1 Artificial Neural Network (ANN)

Figure 3 shows the architecture of an ANN. There is one input layer comprised of a number of nodes. Then there are hidden layers containing several nodes, of which the user can designate the desired number of layers and nodes in each layer. The last layer is the output layer, comprised of nodes which cannot exceed the number in the input layer. Additionally, the output layer should correspond to the number of labels you are trying to predict. Every node in one layer receives an input from all the nodes from the preceding layer, processes that information, and passes the outcome to the following layer (Géron, A, 2019).

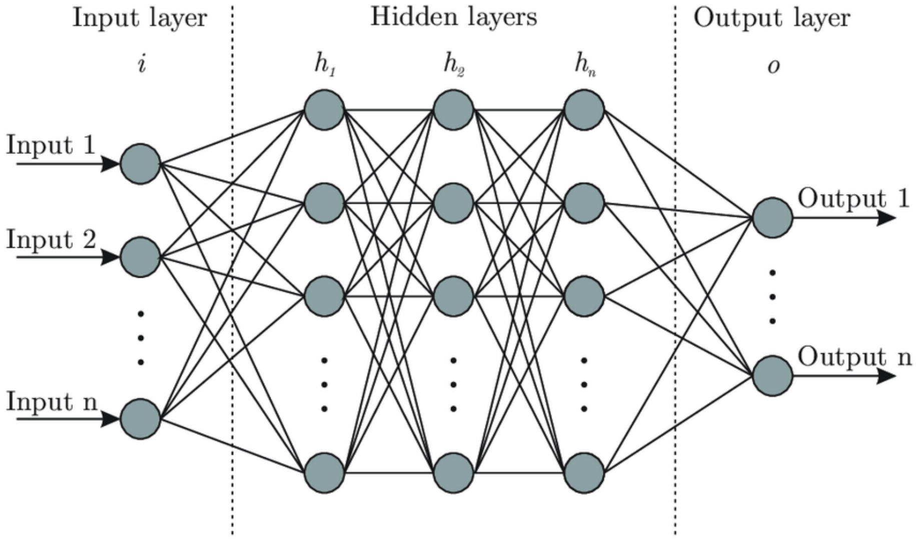


Figure 3. Basic layout of neural network (Source: https://www.researchgate.net/figure/Artificial-neural-network-architecture-ANN-i-h-1-h-2-h-n-o_fig1_321259051)

Figure 4 shows how the ANN algorithm considers each node. Each connection has a weight, and the sum of the variable associated with the node and the weight associated with the connection, along with any bias constant added to the node, results in the node's output. In ANN, the algorithm tries to minimize the objective function, which consists of a loss function, across each iteration. By specifying a loss function, which usually is the mean squared error, the model adjusts these weights to make predictions closer to the true value. Thus, by minimizing the mean squared error between the predicted value and the actual value based on the weight's adjustments, a good predictive model can be achieved.

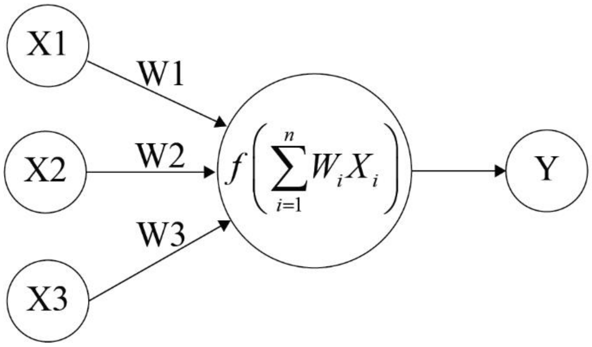


Figure 4. Neural network algorithm (Source: https://subscription.packtpub.com/book/data/9781788397872/1/ch01lvl1sec12/weights-and-biases)

Using the default hyperparameters in the ANN algorithm, the model most often does not produce the best predictions possible. Therefore, these parameters must be adjusted to this specific case, referred to as hyperparameter tuning to develop the best model. More information regarding the specifics of hyperparameter tuning is described in the following subsection.

2.3.2 eXtreme Gradient Boosting (XGBoost)

XGBoost is a decision tree model which utilizes gradient boosting to predict a single output with a set of inputs (XGBoost, 2022a). It is a supervised learning model consisting of an objective function to minimize comprised of a loss term and a regularization term. The loss term predicts the output, and some common functions used for this term are mean squared error and logistic loss. The regularization term deals with the model's complexity to prevent overfitting. Two main types of models can be made: classification and regression. Classification deals with discrete variables while regression deals with continuous variables. Here, a regression model was used.

Figure 5 shows the architecture for an XGBoost model. In XGBoost, the model runs with a training set and gets a prediction, calculating the loss of the prediction compared to the training output. Each leaf has a gradient statistic associated with it, and the set of statistics for the leaves along the path in the tree is used in the prediction through the objective function in the model. XGBoost assigns a weight to wrong predictions to preferentially improve that part of the model. Then it goes back and splits a leaf in two on the tree to try to improve the prediction. If the split improves the model above a set amount, producing enough gain compared to a given constant, it is added to the tree. This is also called a similarity score, meaning that if the new leaf is not similar enough to the old one and instead produces enough gain in the model then it is added. This part of XGBoost's algorithm allows for pruning within the model. XGBoost uses an ensemble model, as illustrated in Figure 6, where one tree is added sequentially to minimize the error in the prediction and then the final prediction score is the sum of the trees' scores. Too deep can lead to overfitting, and after many trees, adding another will not improve the prediction significantly.

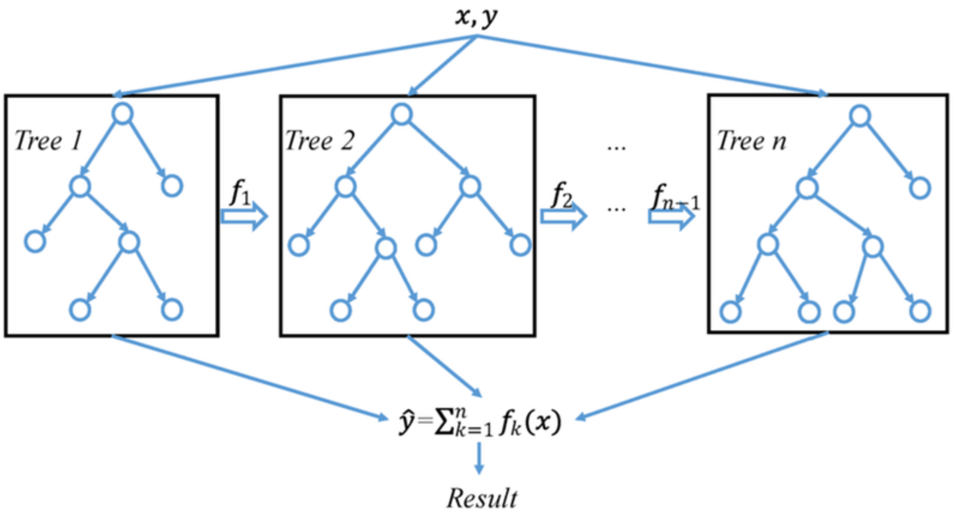


Figure 5. Building an XGBoost model (Source: https://www.researchgate.net/figure/A-general-architecture-of-XGBoost_fig3_335483097)

2.3.3 Hyperparameters

Hyperparameters are variables that help adjust the configuration of the model. Different algorithms use different hyperparameters, but there are some commonalities among hyperparameters across multiple algorithms. The most important hyperparameters regarding ANN models are the hidden layers, number of nodes, and step size. Hidden layers are shown previously in Figure 4 and act as the overall complexity and size of the model. The number of nodes in each hidden layer displays the size and complexity of each layer, and the rate that the model changes is the step size. These hyperparameters help improve the accuracy of the model and help prevent overfitting and underfitting. Overfitting occurs when the model is too complex for the amount of data, which makes it hyperfixate on the training data. This creates extremely high training accuracies but low testing accuracies. Underfitting is when the model is not complex enough for the data and is unable to understand the correlation between the inputs and outputs well enough to generate accurate predictions.

In ANN, the recommended way to tune a neural network is first the hyperparameters and second the layers (Géron, A, 2019, pp. 320-327). Some important hyperparameters include the learning rate, units, batch size, epochs, and activation function. Units refers to the number of neurons within each layer. The batch size designates how many samples are run in a group. Epochs tell how many times the entire dataset is run through the model. The activation function determines which neurons are active in the model. Other examples of hyperparameters include which loss function to use and the dropout rate (the rate that the neural network ignores different layers to create different versions of itself, which are then cross compared to see which is the most accurate).

There are many ways to tune the hyperparameters, some of which include grid search, genetic algorithm, bayesian optimization, and hyperopt. These are all various methods to find the best hyperparameters for the model, and different methods are better for different situations.

Details and results of the models can be found in section 3: Results and Discussion.

3. Results and Discussion

The raw data was imported from an Excel Sheet into a Pandas DataFrame. Some chemicals had missing data for various features, so the missing data was imputed using the K-means clustering algorithm described in Section 2.2. After testing the accuracy of the clusters, the results show the predictions made using the clustering algorithm were more accurate than with simple imputation methods or other algorithms. After filling in the missing data, the top five molecular and top five thermodynamic features were chosen to be used as the input variables in the regression. Over 200 features were included in the data set, collected using the code described previously. The main benefits of narrowing down the number of features used in the model are to simplify the input data needed and to minimize the amount of data users would need to input to use the model for finding the LCI of their chemicals. Two feature selection tools were examined from the sklearn package: SequentialFeatureSelector and SelectFromModel. All statistics and accuracy numbers included in this results section use the SequentialFeatureSelector. Sklearn's StandardScaler was used to scale the data, and the data was split into training, cross validation, and testing sets. The percentages for the split can vary, but the numbers included in this section use a 70/18/12 split, respectively. Additionally, any apparent outliers were removed; specifically, data for climate change, also referred to as global warming potential, greater than 20 kg CO2-eq were removed.

3.1 Machine Learning Code: XGBoost

In XGBoost after splitting the data, the training set was used to fit a regression model with an objective function using squared error. The model then used the cross validation set, called the "eval_set" in XGBoost, to improve the model's fit. Previously, a principal component analysis (PCA) was used to attempt to improve the model, but it did not improve the predictions any of the metrics, as seen in Appendix A, and thus was not included in the final version. In some cases, it even worsened the predictions. The hyperparameters were tuned using the hyperopt package, and their final values are included in the code for each model. The hyperparameters used in the model were n_estimators, learning_rate, max_depth, min_child_weight, subsample, colsample_bytree, and gamma. Verbosity was set to 0 and a random_state of 1 was used. Figure 6 shows the graphs of each of the four LCIs comparing the model's prediction to the true value in the testing set, and the R^2 and root mean squared error are also included in the graphs. Two variations of the code were produced, one organized using functions and the other organized using classes. When organized into classes, two classes were made, one to format the raw data and another to generate the model and analyze the predictions. The file organized using functions includes the most recent code.

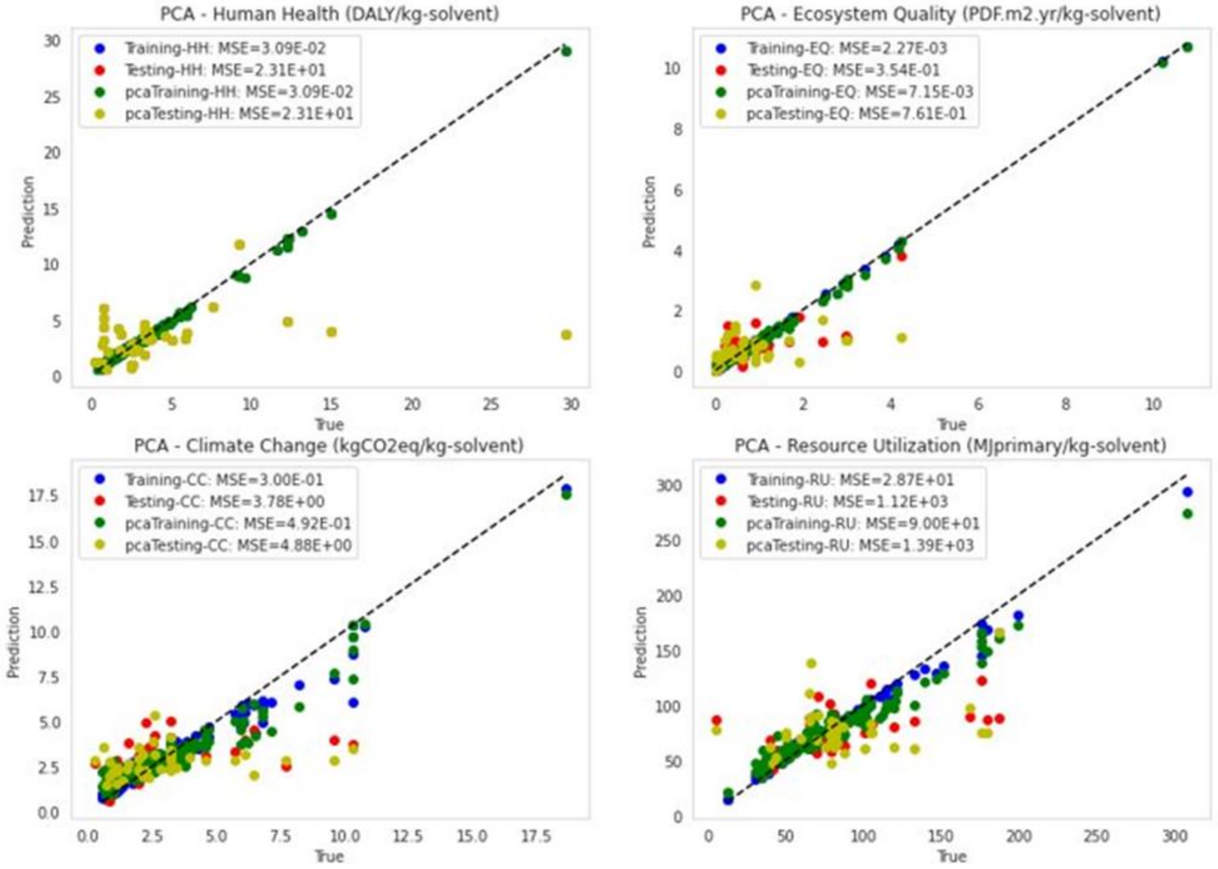


Figure 6. XGBoost model results for all Life Cycle Metrics

The training set run through the model produced predictions very similar to the true values for all four of the LCI output features. The model where the testing set predictions most closely matched the testing set true values was for global warming potential, in which R^2 was 0.727. That and human health, in which R^2 was 0.700, were the best fit models, while ecosystem quality was a good fit with an R^2 of 0.576. The resource utilization's model did not have a good fit, with an R^2 value of only 0.321.

3.2 Machine Learning Code: ANN

In ANN after splitting the data, a model was fitted using the training data and cross validation set. The hyperparameters tuned were learning rate, units, batch size, epochs, activation function, and number of layers. The loss function used was mean squared error. The LCI value of Global Warming Potential was analyzed and shown in Figure 7, which also shows the root mean squared error and R^2 values.

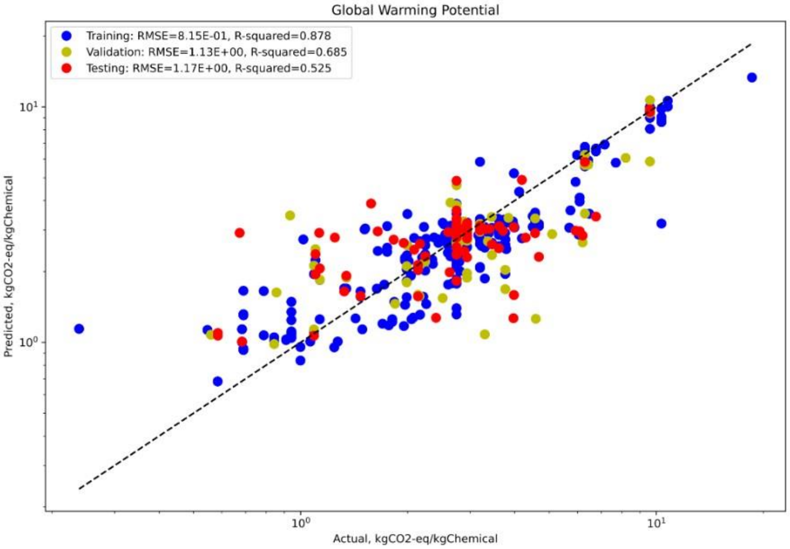


Figure 7. ANN model results for Global Warming Potential

In general, the predictions using the ANN model found some correlation between the 10 properties input and the Global Warming Potential LCI estimation. The R^2 for using ANN's model was 0.525 for the testing set and 0.878 for the training set. The root mean-square error for the testing was 1.17 and 0.878 for the training set. The testing set is much less accurate than the training set using both error methods, meaning that overfitting may be an issue. One other source of error may be from the data itself. With the clusters averaging 16.74% error combined with manual searching and changing of incorrect data, the data may not be entirely accurate to the true LCI values. Additionally, with each increase in total data (including both the number of chemicals and number of properties per chemical), the model's accuracy increases.

3.3 Case Study: Wine Pomace

The LCI data from the machine learning algorithm can be used in conjunction with data from case studies in order to produce a more complete LCA analysis. There are three major life phases within a chemical's lifespan: the production phase, the use phase, and the end-of-life phase. While both the use phase and the end-of-life phase can be modeled using ASPEN and SimaPro, the LCI for the production phase cannot be easily modeled using traditional methods. As a result, the ML algorithm predicts the cradle-to-gate LCI data, which represents the production phase. By employing a case study as shown in Figure 8, it is possible to incorporate all a chemical's life phases to produce a complete picture of the impact it has throughout its entire lifetime. The process selected as a basis for this case study analysis was published by Dr. Croxatto Vega et al. (2021) and describes a process involving the recovery of polyphenols from wine pomace. This process involved the grinding of wine pomace followed by the extraction of the polyphenols using a solvent comprised of 67%wt acetone. Following the extraction, the spent pomace was pressed and desolventized, with the vapors from the desolventizer entering a condenser. Meanwhile, the liquid stream from the presser entered a distillation column, where the acetone was separated from the polyphenols dissolved in water and was recycled along with the liquid from the condenser. The water-polyphenol mixture from the distillation column then entered both a nano filter and a spray dryer, with the mostly dry polyphenols serving as the final product for the process. Throughout the process, it is estimated that 2% of the acetone solvent is lost each cycle, requiring some solvent to be replenished each cycle.

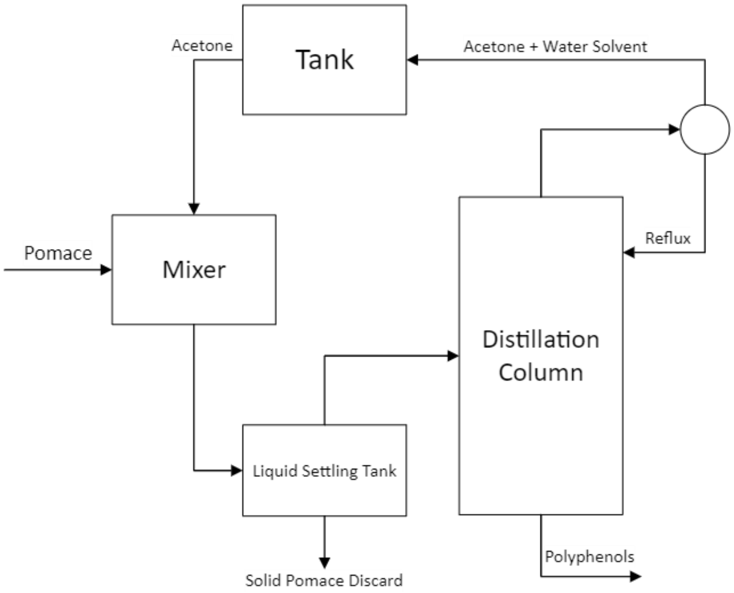


Figure 8: Process Flow Diagram for Extraction of Polyphenols from Wine Pomace

To compare the data reported in the paper with the LCI data generated by the machine learning model, the basis of calculation for the paper's energy and mass input data was converted from polyphenol to acetone. The energy and mass input data were then corroborated with data from SimaPro and Aspen to calculate the gate-to-gate LCI values. These gate-to-gate LCI values represent the impact the polyphenol extraction process had in its entirety, serving as a use case scenario for acetone. Conversely, the LCI data from the machine learning algorithm was a cradle-to-gate analysis, serving as the production phase for acetone. The gate-to-gate global warming potential (GWP) derived from the paper is 15.0 kg CO_2/kg Acetone, while the cradle-to-gate GWP derived from the machine learning algorithm is 1.90 kg CO_2/kg Acetone. Additionally, since the only acetone leaving the process is through fugitive emissions, the end-of-life gate-to-grave GWP can be assumed to be 0 kg CO_2/kg Acetone. By using all of these values in conjunction, it is possible to complete a full cradle-to-gate LCA for acetone, with a GWP of 16.9 kg CO_2/kg Acetone. This value paints a clearer picture of the impact that the acetone solvent has throughout its entire life span by being able to represent all three of its major life phases.

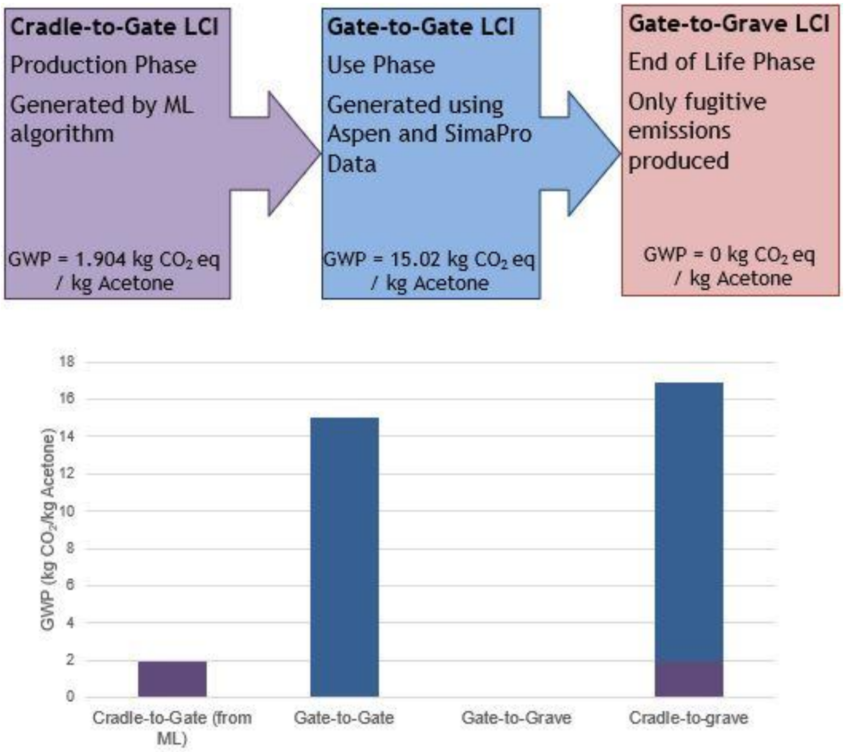


Figure 9: Complete LCA Analysis Results

4. Conclusions and Future Work

In this study, the Machine Learning algorithms, XGBoost and Advanced Neural Networks (ANN), were used to create a program that predicts the environmental impacts of chemicals in the areas of human health, ecosystem quality, global warming potential, and resource utilization with the goal of creating a tool that can be used to predict the effects of novel chemicals. Data was collected from a variety of chemical databases including Ecoinvent and DIPPR, and missing data was imputed using K-means clustering. The data was collected in an automated fashion, allowing for the collection of copious amounts of data, creating a more accurate and reliable model. Over 200 molecular descriptors and 23 thermodynamic properties were analyzed, and a stepwise feature selection process was carried out to reduce the number of features from 223 to 10. An analysis of these properties was carried out to predict chemicals' global warming potential (GWP, R^2 =0.727), human health impact (HHI, R^2 = 0.700), ecosystem quality impact (EQI, R^2 =0.576), and resource utilization impact (RUI, R^2 =0.321) using XGBoost and global warming potential (R^2 = 0.53) using ANN. Based on these results, the XGBoost algorithm provides a better prediction of the global warming potential of chemicals. An ANN algorithm will be developed and used to predict the HHI, EQI, and RUI of chemicals and the results of these studies will be compared to the predictions from XGBoost.

A case study of acetone in wine pomace was also carried out to demonstrate the utility of the ML tool. This case study showed that the algorithm's predictions can be used to substitute unknown data: where the cradle-to-gate LCI information was not available for acetone, the ML algorithm was able to fill in the gaps and predict this information, allowing for a complete cradle-to-grave analysis when combined with other known data for the gate-to-grave analysis.

The models will continue to be improved through the extraction of more data, which can be achieved using databases like Ecoinvent that allow for efficient collection through automation. More data will allow the model to use chemical properties to find a better correlation, which will be demonstrated by an R^2 closer to 1. Further work with this model includes the creation of emissions scaling equations that would allow for a better prediction of the effects of chemicals' large-scale production on the environment for a variety of different chemical processes.

A web-based graphical user interface (GUI) will also be created that would allow user-friendly access to the Machine Learning data and results outlined by this report. The tool would allow users to predict the environmental impacts of novel chemicals they are developing or producing, reducing the research and laboratory work required, thereby reducing the overall economic and environmental cost of process development.

5. Acknowledgments

This research has been supported by a grant from the American Institute of Chemical Engineers' Sustainable Engineering Corps Student Projects funded by the United Engineering Foundation. The assistance and guidance provided by the Department of Chemical Engineering at Rowan University, Dr. Kirti Yenkie, Dr. Robert Hesketh, Emmanuel Aboagye, and Austin Lehr throughout the program was essential to the product as well and is greatly appreciated.

6. References

Aboagye, E. A., Chea, J. D., & Yenkie, K. M. (2021). Systems level roadmap for solvent recovery and reuse in industries. *iScience*, *24*(10), 103114–103114. https://doi.org/10.1016/j.isci.2021.103114

Argoti, A., Orjuela, A., & Narváez, P. C. (2019). Challenges and opportunities in assessing sustainability during chemical process design. *Current Opinion in Chemical Engineering*, *26*, 96–103. https://doi.org/10.1016/j.coche.2019.09.003

Calvo-Serrano, R., González-Miquel, M., Papadokonstantakis, S., & Guillén-Gosálbez, G. (2018). Predicting the cradle-to-gate environmental impact of chemicals from molecular descriptors and thermodynamic properties via mixed-integer programming. Computers & Chemical Engineering, 108, 179–193. https://doi.org/10.1016/j.compchemeng.2017.09.010

Croxatto Vega, G., Sohn, J., Voogt, J., Birkved, M., Olsen, S. I., & Nilsson, A. E. (2021). Insights from combining techno-economic and life cycle assessment – a case study of polyphenol extraction from red wine pomace. Resources, Conservation and Recycling, 167, 105318–. https://doi.org/10.1016/j.resconrec.2020.105318

Finnveden, G., Hauschild, M. Z., Ekvall, T., Guinée, J., Heijungs, R., Hellweg, S., Koehler, A., Pennington, D., & Suh, S. (2009). Recent developments in Life Cycle Assessment. Journal of Environmental Management, 91(1), 1–21. https://doi.org/10.1016/j.jenvman.2009.06.018

Géron, A. (2019). *Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (2nd ed.). O'Reilly Media.

Karka, P., Papadokonstantakis, S., & Kokossis, A. (2019). Predictive LCA - a systems approach to integrate LCA decisions ahead of design. In A. A. Kiss, E. Zondervan, R. Lakerveld, & L. Özkan (Eds.), *Computer Aided Chemical Engineeri ng* (Vol. 46, pp. 97–102). Elsevier. https://doi.org/10.1016/B978-0-12-818634-3.50017-5

Mercado, G. R., & Cabezas, H. (2016). *Sustainability in the Design, Synthesis and Analysis of Chemical Engineering Processes*. Butterworth-Heinemann.

Narciso, D. A. C. & Martins, F. G. (2020). Application of machine learning tools for energy efficiency in industry: A review. Energy Reports, 6, 1181–1199. https://doi.org/10.1016/j.egyr.2020.04.035

Nguyen, Diaz-Rainey, I., & Kuruppuarachchi, D. (2021). Predicting corporate carbon footprints for climate finance risk analyses: A machine learning approach. Energy Economics, 95, 105129. https://doi.org/10.1016/j.eneco.2021.105129

Sala, S., Ciuffo, B., & Nijkamp, P. (2015). A systemic framework for sustainability assessment. *Ecological Economics*, *119*, 314–325. https://doi.org/10.1016/j.ecolecon.2015.09.015

Sheldon, A.R. (2007). The E Factor: Fifteen years on. *Green Chemistry, 9*(12), 1273-1283. https://doi.org/10.1039/B713736M

UN Environment Programme. (2019). *Global Chemicals Outlook II-- From Legacies to Innovative Solutions: Implementing the 2030 Agenda for Sustainable Development*. https://www.unep.org/resources/report/global-chemicals-outlook-ii-legacies-innovative-solutions?_ga=2.52250473.1764953922.1683398988-439696597.1683398988

US EPA O (2013) TRI Data and Tools. In: US EPA. https://www.epa.gov/toxics-release-inventory-tri-program/tri-data-and-tools. Accessed 30 Jul 2018

XGBoost. (2022a). *Introduction to Boosted Trees*. XGBoost Documentation.
https://xgboost.readthedocs.io/en/stable/tutorials/model.html.

XGBoost. (2022b). *XGBoost Parameters*. XGBoost Documentation. https://xgboost.readthedocs.io/en/stable/parameter.html