

# Development of a Catalysis Analytics Platform

## Enabling Machine Learning in Catalyst Discovery

Majid Mohamedhoesein

Technische Universiteit Delft

# DEVELOPMENT OF A CATALYSIS ANALYTICS PLATFORM

## ENABLING MACHINE LEARNING IN CATALYST DISCOVERY

by

**Majid Mohamedhosein**

in partial fulfillment of the requirements for the degree of

**Master of Science**

in Chemical Engineering

at the Delft University of Technology,

to be defended publicly on 22nd of April 2021.

Student number: 4629604

Supervisors: Prof. Dr. Evgeny Pidko  
MSc. Robbert van Putten

Thesis committee: Prof. Dr Evgeny Pidko  
Dr. Atul Bansode  
Dr. Artur Schweidtmann

# ABSTRACT

Recently machine learning (ML) has become increasingly popular, and has been shown to be a powerful predictive technique. The applications of ML cover a wide range of disciplines, including the natural sciences. Presently, the field of catalysis is still relatively unexposed to ML and other data-driven techniques. This can largely be attributed to the broad variety and complexity of catalytic data, which obstructs data unification into large structured databases. This is problematic because ML requires large amounts of information rich data to ensure its effectiveness. Additionally, applying ML techniques requires expertise and often coding experience. These requisites impede the adoption of ML by catalysis researchers, that are not necessarily programming experts. In this thesis a data management and analytics platform is developed to reduce this inaccessibility barrier. Our platform is designed to guide catalysis researchers through the ML workflow, and construct effective ML models. Throughout this process the platform supports several key functionalities, which include interaction with a database instance, data visualization, ML model construction, and ML model application.

Furthermore, the usefulness of our platform is demonstrated in a case study, where ML models are built to predict the catalytic performance based on molecular descriptors of the catalyst. Due to a lack of suitable existing catalytic datasets, we construct and use an artificial dataset that mimics kinetic catalytic data. Artificially constructing the data allows us full control of its underlying mechanisms. This aspect is used to build dataset variations where we study the effect of database size and descriptor strength on the performance of ML models. The results of the case study quantify the effects of database size and descriptor strength, which are useful in identifying the objectives for future construction of databases from real catalytic data.

# CONTENTS

<b>Abstract</b>	<b>i</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Methodology shift in catalysis research	2
1.2 Catalysis informatics	3
1.3 Thesis objectives & outline	4
<b>2 Foundations of ML</b>	<b>6</b>
2.1 Fundamentals	6
2.2 ML workflow	7
2.2.1 Data collection	8
2.2.2 Data exploration & preparation	8
2.2.3 Algorithm selection	9
2.2.4 Model training	9
<b>3 Platform development</b>	<b>10</b>
3.1 Platform infrastructure	10
3.2 Electron	11
3.3 Database management	12
3.3.1 Database structure	12
3.3.2 MongoDB	13
3.3.3 CRUD operations	14
3.4 Data visualization	17
3.5 Modelling	18
3.5.1 AutoML	19
<b>4 Case study</b>	<b>24</b>
4.1 Dataset construction	25
4.1.1 Kinetic model	26
4.1.2 Activation energy correlation	28
4.1.3 Dataset variations	29
4.1.4 Evaluation method	31

---

4.2 Results & Discussion . . . . .	34
<b>5 Conclusion</b>	<b>47</b>
5.1 Outlook . . . . .	48
<b>6 Acknowledgements</b>	<b>49</b>
<b>Bibliography</b>	<b>50</b>

# 1

## INTRODUCTION

The influence of machine learning (ML) is greater than ever, and increasing its scope. Nowadays, ML has impacted a wide range of disciplines. For instance, applications of ML include human face detection,<sup>1</sup> stock price forecasting,<sup>2</sup> and adding self-driving features in cars.<sup>3</sup> ML also influences the natural sciences. In late November 2020, Deepmind, one of Google's artificial intelligence companies, used ML to tackle the notorious 50-year-old challenge that is the deduction of the 3-dimensional shape of proteins from their amino-acid sequence. Proteins are molecules made out of a collection of amino-acids that fold into a conformation of lowest energy. Such proteins are made of very long chains of amino-acids. Molecular interactions between these amino acids, and their place in the chain result in specific folding of the protein that is critical for its function. The immensity of the possible conformation space has made predicting the intricate 3D structure of a protein a large challenge in biological research.<sup>4</sup> Deepmind's approach to the problem trains an artificial neural network (ANN) on publicly available data of 170,000 protein structures. Deepmind has shown that their ML approach to the problem resulted in a model of unprecedented accuracy, which left many scientists to believe that the problem was essentially solved.<sup>5</sup>

Over the years ML has helped achieve similar breakthroughs in other scientific fields. However, one field comparably unexposed to machine learning is catalysis. This is unexpected given the similarity between catalysis and biology's protein folding challenge, in the sense that both fields cope with complex interactions and a vast space of possible chemical structures. However, catalysis has a unique additional layer of complexity, which arises from its transient and dynamic nature. Generally, the process of catalysis proceeds via sequences of steps, such as diffusion or the binding and reacting of substrates at the catalyst's active site. These steps are affected by a multitude of factors, and all have their own length and time scale. Moreover, catalysis knows many research methods and metrics for quantifying the dynamic catalytic behaviour. As a result, catalysis data frequently lacks in data uniformity, which obstructs the formation of large structured databases.<sup>6</sup> For example, the catalytic metric "activity" can alter in definition in different scientific works.<sup>7</sup> In this regard, catalysis can benefit from

incorporating ontology, which is a formal way of representing knowledge in the form of concepts that belong within a given domain.<sup>8</sup> The development of ontologies is complex, and is not addressed in this thesis project.

Currently, the absence of large structured databases is a barrier in the adoption of data-driven research techniques in catalysis, and several challenges remain to be addressed. In this work we aim to promote the adoption of data-driven research techniques within catalysis by development of an analytics tool. Additionally, this tool will be validated by means of a case study. Before introducing the central challenge of this thesis it is important to discuss the current state of the catalysis field and where data-driven techniques can contribute to more effective research.

## 1.1. METHODOLOGY SHIFT IN CATALYSIS RESEARCH

Within the large field of catalysis, the objectives of this research area can principally be condensed into a single goal. This goal is to understand how to design catalyst structures to control catalytic activity and selectivity.<sup>9</sup> The development of desirable catalysts is integral to tackling current and future challenges, such as producing clean energy and creating safe pharmaceuticals.

The traditional approach in catalysis is to form theories that explain the inner workings of catalytic behaviour. These theories are formed by identifying cause-and-effect relations and constructing hypotheses that are based on past observations and intuitions. Subsequently, the viability of the hypotheses is tested by experiments or simulations. This method of research is referred to as the theory-driven approach. Over time, considerable understanding and insight has been gathered as a result of the theory-driven approach. However, there are shortcomings attached to this method of research.

The shortcomings of the theory-driven approach can largely be attributed to the complexity of the chemistry in catalysis. The observed phenomena that catalysis aims to describe and understand are usually dependent on many variables. Considering this fundamental characteristic of catalysis that "everything depends on everything", building theories where all relations between the variables and the catalytic behaviour are determined is an impractical task for complex catalytic systems. For instance, including reaction condition dependency in DFT calculations remains a large challenge for many catalytic systems. Therefore, the models produced by the theory-driven approach often inadequately predict the observed behavior of complex catalytic systems.

Predictive capabilities in accurately predict catalytic properties is of great value when deciding what experiments should be conducted. The space of possible catalysts is close to boundless. An upper estimate of the magnitude of chemical space says it contains  $10^{180}$  compounds, which is more than twice the magnitude of the number of atoms in the universe.<sup>10</sup> Being able to predict which catalysts in this large chemical space will exhibit desirable behaviour, has the potential to reduce the number of

necessary experiments. Consequently, accurately predicting catalyst performance and properties leads to a more efficient workflow. The result is a research workflow with faster development of effective catalysts, while using less resources.

The shortcomings of traditional research method are complemented by the modern data-driven approach, which includes data-driven technologies in scientific research. This method of research has become increasingly popular and powerful as a result of recent advances in availability of standardized data and computing power. Predicting the outer behaviour of catalysis is specifically addressed by data-driven technologies. The observed catalytic behaviour is captured in the form of data, which forms the basis for predictive tools, such as ML. The predictions of ML are solely based on the provided data, which means that, opposed to the theory-driven approach, minimal intuition is involved. This makes the data-driven approach, in a sense, unbiased and primarily dependent on the quality of the data. Additionally, excluding human intuition from the decision making process in catalysis contributes to the automation of catalytic research.

Although, catalytic research that includes the modern data-driven approach is still relatively young, there have been efforts in the development and implementation of data-driven techniques in catalysis. These efforts are part of the relatively new field, which is labeled catalysis informatics.

## 1.2. CATALYSIS INFORMATICS

The area of catalysis informatics involves research at the interface of information science and catalysis. The fusion of a particular scientific discipline with information sciences has occurred several times before. Previously, chemistry, biology, and material science have made similar transformations, which respectively gave rise to the fields of cheminformatics, bio-informatics and materials informatics. The emergence of these fields has since resulted in the development of influential data-driven tools and systems. In the domains of cheminformatics and material informatics these involve the development of retrosynthetic planning<sup>11</sup> and quantitative structure–activity/property relationship (QSAR/QSPR) models.<sup>12</sup> The availability of large volume datasets contributed significantly to the development of these data-driven tools, where combined databases could comprise over half a million different compounds.<sup>13</sup>

In catalysis informatics, comparable large volume datasets are currently not available. However, smaller datasets have been used in the identification of catalytic descriptors and scaling relationships.<sup>14,15</sup> In addition, data-driven techniques have been used in accelerating computationally intensive electronic structure calculations by composite methods that include ML and density functional theory (DFT).<sup>16,17</sup> These efforts have mostly been made in specific subsections of catalysis, such as surface adsorption reactions.



Moreover, catalysis informatics has recognized the absence of large volume structured databases.<sup>18</sup> Despite the large number of published results the dataset size rarely exceeds the order of thousands. Intended to construct a large volume catalysis database, the natural inclination of combining different literature sources jeopardizes dataset quality. The pitfalls are commonly found in lack of overlap between data sources and bias towards well-performing catalysts.<sup>19,20</sup> The latter phenomenon originates from literature's inclination towards reporting successful experiments.

Although there are currently no catalytic databases where multiple data sources are combined into a single dataset, web-based data platforms<sup>21,22</sup> have been developed to promote centralized data storage and data accessibility. For example, the Catalysis Hub contains computationally calculated reaction energies and barriers of thousands of surface reactions. At the moment of writing, Catalysis Hub holds reactions and DFT structures from 69 different publications.

While these platforms contribute to efficiently accessing meaningful data, using this data in cohesion with analytic and data mining techniques is not addressed by these platforms. The integration of these techniques into a platform is crucial to promote the adoption of the data-driven approach by catalysis researchers. These researchers are usually unfamiliar with the softwares or programming languages that are typically used in the ML process. Additionally, many steps in the ML workflow require great expertise in order to construct effective models. This is problematic because catalysis researchers generally do not possess of this data science and ML expertise. Consequently, catalysis researchers who want to include data-driven techniques into their research are obstructed by an adoption barrier. We define this as the inaccessibility problem.

It is worth mentioning that a more recent approach in platform development recognized the inaccessibility problem and included options for analysis through data visualization in their platform.<sup>18</sup> Although data visualization is a central tool in the process of ML, there is currently no platform that supports catalysis researchers in the construction of new ML models. Thus, valuable insights and opportunities are left to be exploited

### 1.3. THESIS OBJECTIVES & OUTLINE

In this thesis a platform was developed capable of providing database interaction and guiding the catalysis researcher through the entire ML workflow. Our platform supports database management, data visualization, the construction and application of predictive ML models. Moreover, the platform offers these functionalities through an intuitive graphical user interface (GUI), which disregards the need for prior coding experience.

Next, the functionality of the platform is demonstrated through a case study. Due to the lack of an appropriate available dataset, the dataset used in the case study is artificially constructed, and intends

to mimic future catalytic datasets. In addition to evaluating the platform's performance, the case study serves as a proof of concept, and aims to answer questions regarding the potential of future catalysis datasets.

The next chapter discusses the foundations of ML that are relevant to this work. Chapter 3 explains the architecture and development of the database and analytics platform. The fourth chapter discusses the case study, which involves the construction of the artificial dataset, as well as the platform's performance in ML model construction. Finally, the conclusions are gathered in the fifth chapter.

# 2

## FOUNDATIONS OF ML

### 2.1. FUNDAMENTALS

Machine learning (ML) is a branch of the field of artificial intelligence (AI) where models are produced by the automated detection of meaningful patterns in data. The procedures used to construct the ML models are called ML algorithms. These algorithms are provided with input data along with desired outcomes and adapt their architecture through mathematical optimization techniques, such as back-propagation and gradient descent, to improve performance on the desired task.<sup>23</sup> This process of adaptation is called training. Model training results in a model that has learned to achieve a desired task from the provided data. An important characteristic of ML is its ability to generalize the training experience, which allows the constructed model to perform well on unseen data.<sup>24</sup>

Most ML algorithms can be divided into two categories: supervised and unsupervised learning. Supervised learning is used to predict the value of a target variable from a set of input variables that are called features. Unsupervised learning is generally used to detect clusters in datasets based on similarity. In this work we will focus on supervised learning.

Within supervised learning further distinction can be made between classification and regression. These two predictive modelling problems differ in the nature of the target variable. Fundamentally, classification models have discrete target variables, whereas regression models have continuous target variables. To illustrate, consider the task of deciding the shade of an image based on certain features. As can be seen in figure 2.1, the exemplary classification model only outputs discrete labels, either black or white. In contrast, the exemplary regression model outputs a numerical value on the continuous gray scale spectrum.

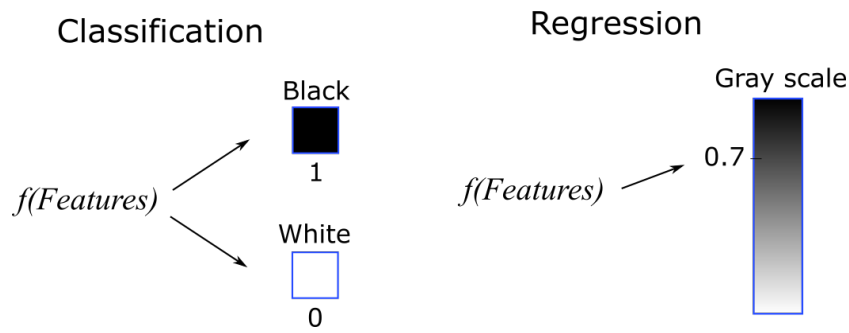


Figure 2.1: Example of a classification and regression ML model

## 2.2. ML WORKFLOW

Generally, the process of producing ML models consists of 6 steps. As the ML workflow is integral to the approach of designing the analytics platform, this section serves to explain the individual steps of the ML workflow. A full overview of the ML workflow is shown in figure 2.2.

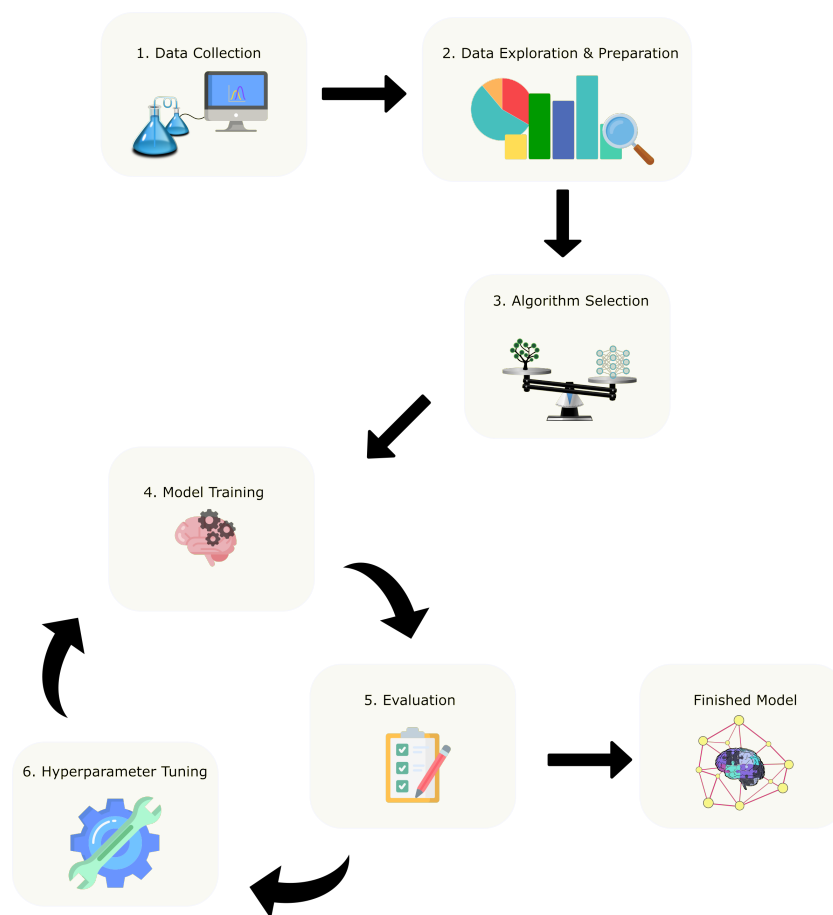


Figure 2.2: An overview of the machine learning workflow

### 2.2.1. DATA COLLECTION

As ML is an inherently data-driven technique, the workflow starts with the collection of data. In the context of this thesis the data is collected from catalysis experiments and simulations. For example, automated electronic structure calculation and high-throughput experimentation are promising methods for the collection of large quantities of catalytic data.<sup>19</sup> Generally, the resulting data is collected in a file, for instance a Microsoft Excel file.

### 2.2.2. DATA EXPLORATION & PREPARATION

"Garbage-In-Garbage-Out" is a well-known saying in ML about how the output of a ML model can only be as good as the quality of its input. Therefore, the next step in the ML workflow aims to manage the quality of the dataset. This is done by a combination of data exploration, which aims to understand the patterns in the data, and data cleaning, which aims to transform or even remove problematic data points. A common cause known to impede the construction of valuable ML models is disproportionate data, which is commonly referred to as class imbalances. Class imbalances occur when the distribution of variables is highly biased or skewed. This bias can give rise to false assumptions during the training process. Therefore, class imbalances will often result in ML models that are systematically prejudiced in their predictions. For example, consider a highly imbalanced dataset, where catalysts are labeled either poor or well-performing. If 99% of the catalysts are well-performing, a flawed ML model that blindly labels all inputs as well-performing still achieves a high accuracy of 99%. This discourages the algorithm to find true relationships in the data, because it is unlikely that they lead to better performing models.

In order to detect class imbalances and other relevant relationships, data visualization is used. The process of understanding the data through visualizations is called exploratory data analysis (EDA) and is considered a fundamental part of the ML process. For example this could involve creating histograms that show the distribution of a certain variable, or scatter plots that expose outliers in the data.

After the quality assuring analysis the data is termed preprocessed. The final part of the preparation consists of splitting the data in a train set and a test set. The purpose of the test set is to evaluate the trained ML model's performance. Therefore, the test set should be representative of new data that would be given to the finished model. Generally, increasing the test set fraction achieves higher similarity between the train and test set. However, decreasing the size of the train set could deteriorate the model's performance. Therefore attention should be given to balancing these effects and finding the appropriate split fraction.

### 2.2.3. ALGORITHM SELECTION

ML knows many different algorithms, and selecting the most suitable one requires consideration of many factors. This involves knowing the strengths of the algorithms but also knowing their deficiencies. The Naïve Bayes algorithm for example assumes that all features are independent from one another. When this is not the case, the Naïve Bayes algorithm is most likely not the right choice.

### 2.2.4. MODEL TRAINING

In this section we discuss the final 3 steps of the ML workflow: model training, evaluation and hyperparameter tuning. This sequence of steps is often repeated to increase the model's performance. Essentially, the training process is a cycle of repeatedly adjusting model parameters until convergence or until a set limit of iterations is reached. There are two kinds of these parameters: internal and external parameters. Internal parameters are constants inside of the model that are fitted to the data. This fitting process is performed by an optimization algorithm, that is able to efficiently search for possible parameter values. For example, the coefficients of the well-known linear regression algorithm are internal parameters, and are found by minimizing the sum of squared errors. In contrast, external parameters, which are more commonly known as hyperparameters, are not determined from the provided data and can be manually set by the model practitioner. An example of a model's hyperparameter is the number of neuron layers in an artificial neural network. The external parameters can be viewed as defining a specific model configuration, whose internal parameters will be adjusted such that the model best represents the data. Finally, the model is evaluated by its performance on the test set. At this point the ML model is finished and the evaluation aids to determine whether it can be deemed good enough for the intended cause.

## PLATFORM DEVELOPMENT

### 3.1. PLATFORM INFRASTRUCTURE

In this work a data management and analytics platform was developed. Our platform has the following four main functionalities: database interaction, data visualization, ML model construction, and ML model application. A suitable hardware infrastructure was required to provide these services. An overview of the developed infrastructure is provided in figure 3.1. The researcher's local machine is the starting point of the infrastructure, and is used to access the platform. In addition, the infrastructure involves three servers that each have their own purpose: (1) facilitating the data transfer between all of the involved machines, (2) hosting of a database management system, and (3) providing computational resources. The first feature is necessary because the servers are deployed in TU Delft's data center and are secured with an institutional firewall. This is a key safety measure to ensure confidentiality of the involved data. The functionality of the remaining two servers will be discussed in the upcoming sections.

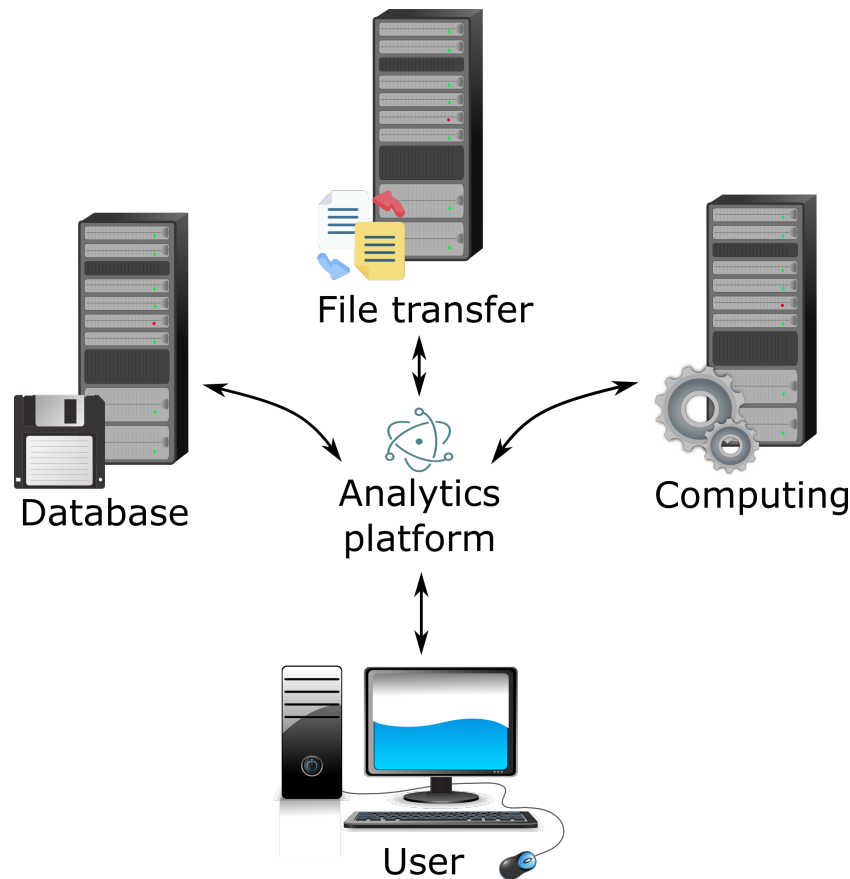


Figure 3.1: A global overview of the platform's hardware infrastructure

Next to the discussed hardware infrastructure, the platform makes use of several softwares to provide its functionalities.

## 3.2. ELECTRON

At the foundation of the platform all of the desired functionalities are brought together under one underlying framework. This is achieved by constructing the platform in Electron. Electron is a framework for creating native desktop applications with web technologies involving JavaScript, HTML, and CSS.<sup>25</sup> Such a desktop environment was necessary because it allows for the integration of external programs. In this work we made use of this feature by incorporating a data visualization program into the platform. The dominant factor in selecting this specific binding framework was Electron's technological maturity and reliability. Many well-known desktop applications have successfully been built in Electron, such as Whatsapp, VS Code, Microsoft Teams and Twitch.<sup>25</sup>

Additionally, the Electron framework manages the graphical user interface (GUI). Upon starting the platform, the home page (shown in figure 3.2) becomes visible. The home page contains five buttons



that each link to a different page. The function and design of these pages will be discussed throughout this chapter.

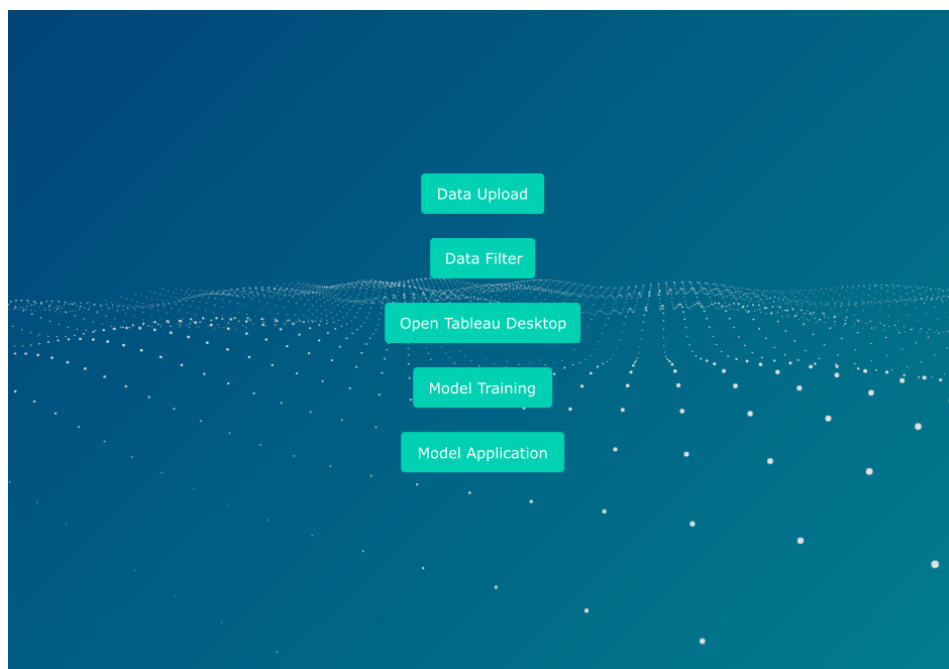
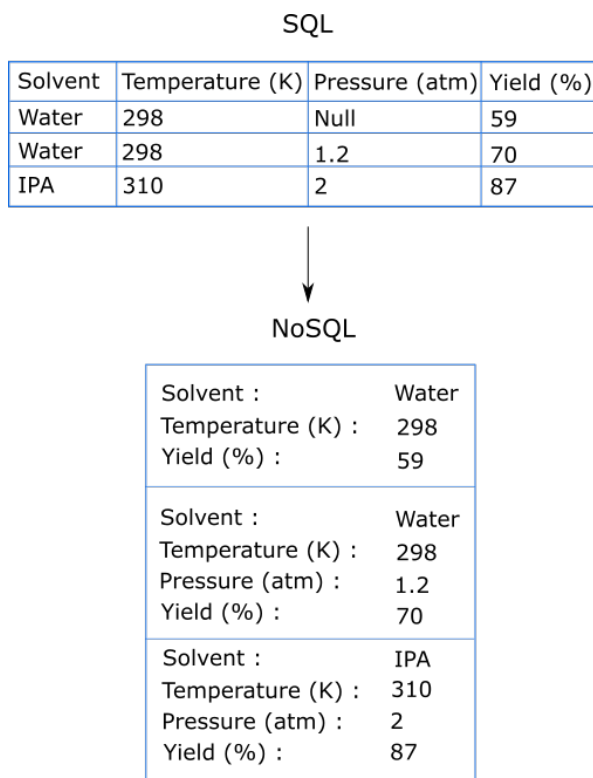


Figure 3.2: Platform page 1: Home screen

### 3.3. DATABASE MANAGEMENT

#### 3.3.1. DATABASE STRUCTURE

The integration of a database management application has previously been explored in our group.<sup>26</sup> Among its findings was the conclusion that a NoSQL (not only structured query language) database is the appropriate database structure for storing catalysis data. Next to NoSQL, there is also the more common SQL (structured query language) type database. SQL databases store data in tabular form with labelled rows and columns and is widely known from spreadsheet software, such as Microsoft Excel. The drawback arising from this method of data storage is that data is forced to adhere to the predefined tabular structure. This is troublesome because catalysis data is diverse, and the relevant variables could vary across catalysis research. Opposed to SQL, NoSQL does not force data to follow the rigid structure of columns and rows. This is illustrated in figure 3.3 where the same data is presented in both SQL and NoSQL format. NoSQL's flexibility towards different data structures was the decisive factor in choosing the database structure because it allows various formats for catalytic data to be effortlessly stored in the database.



**Figure 3.3:** A comparison between storing data in a SQL and NoSQL format

### 3.3.2. MONGODB

As previously shown in figure 3.1 a server is employed for hosting the NoSQL structured database. This is achieved by equipping the server with a database management system (DBMS). MongoDB was the selected DBMS of choice for a number of reasons: MongoDB is open-source, can be self-hosted on private servers, and is able to store arrays-in-arrays. The latter refers to storing entire arrays of data in a single input field. This means that it is possible to store for example chromatography or spectroscopy data alongside simple variables such as reaction conditions in catalysis datasets. These reasons along with MongoDB's reputation and widespread adoption by over 4000 companies led to MongoDB being the chosen DBMS<sup>27</sup>

The storage system of MongoDB consists of databases and collections. Databases serve as folders in which multiple collections can be stored. These collections are the actual data that gets uploaded to MongoDB. Before the MongoDB instance can be accessed, a connection to the platform should be established. This is achieved by means of a connection string, which serves a purpose similar of that of a password, and allows for a secure database. This is important because of the potential confidentiality of the stored data.

With the database software installed and connected, we shift focus to the functionalities that are of-

ferred by integration of MongoDB. More specifically, MongoDB provides the platform with the CRUD operations.

### 3.3.3. CRUD OPERATIONS

The core functionalities for effective database storage are commonly summarized by the acronym CRUD (Create, Read, Update, Delete). Between these functionalities the Read operation is the most intricate operation from the user's perspective. In order to avoid unnecessary complexity in the design of the platform's pages, the Read operation has been given its own page. The integration of the four operations over these pages is explained in this section.

The first page is referred to as the Upload page and can be seen in figure 3.4. In this page, databases and collections become available after the connection with MongoDB has been established, and will be shown in the boxes on the left side of the page. When selecting one of the collections, its variables are displayed in the top center box. Showing the collections makes the researcher aware of its contents and data format without extracting the entire dataset from the database. With this information the user can add new data to MongoDB by using the box on the right-hand side of the page. Here we can distinguish between adding data to existing (Update operation) and new (Create operation) databases or collections. The final box in the bottom center of the page can be used to remove collections or even entire databases (Delete operation).

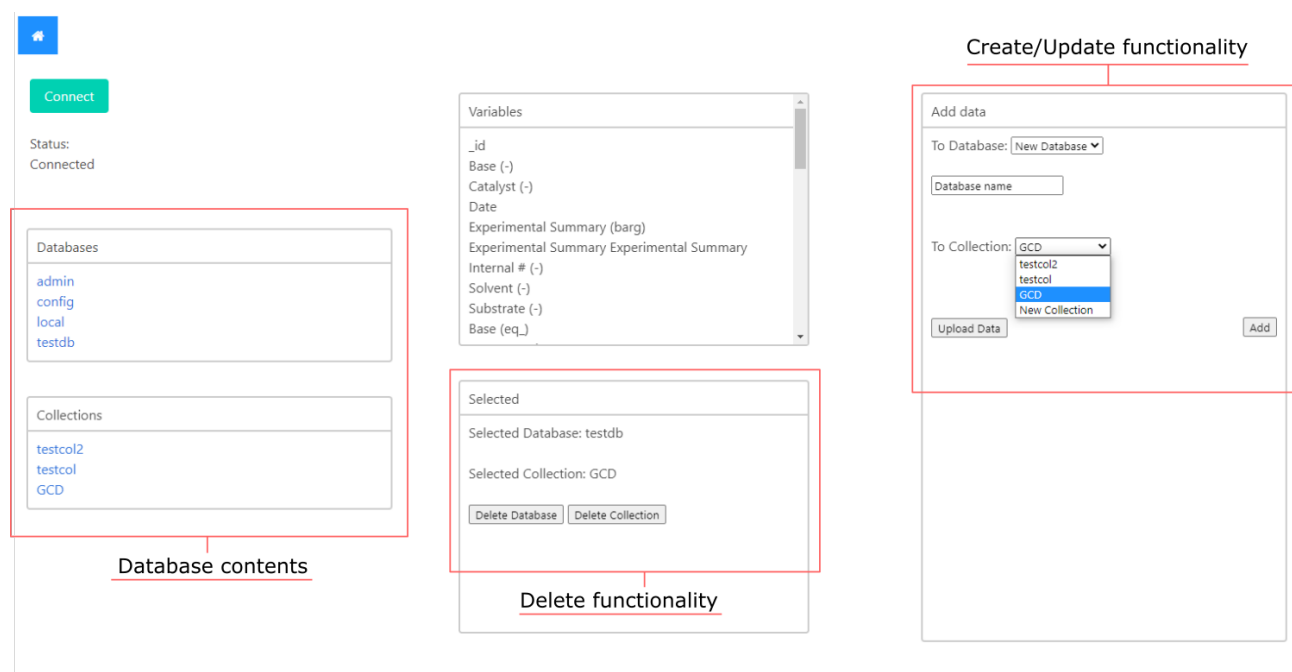


Figure 3.4: Platform page 2: Upload screen

The final CRUD operation (Read operation) is implemented in the filter page (figure 3.5) This page is

specifically designed to retrieve subsets of data from the MongoDB database. With the prospect of a larger databases in the future comes the increasingly difficult and time consuming task of finding the desired data. An effective filter page is crucial to assist the user in efficiently retrieving the targeted data from the database. Similar to the Update page, the databases and collections are loaded inside the boxes on the left side of the page. The remainder of the page is designed to specify what data should be retrieved.

MongoDB is informed of the user's requirements for the retrieved data by a database query, which is a request that is written in a coding language that the database can understand. The filter page allows the user to construct the query without knowing the query coding language. This results in an accessible interface for database communication.

The query is constructed through the following process. First, the user declares which variables should be excluded or retrieved inside of the "Variables" box. The more intricate variable conditions are added by applying filters. The variables to which the filters should be applied are selected in the box on the right-hand side of the page. Finally, the conditions can be specified inside the box at the bottom of the page.

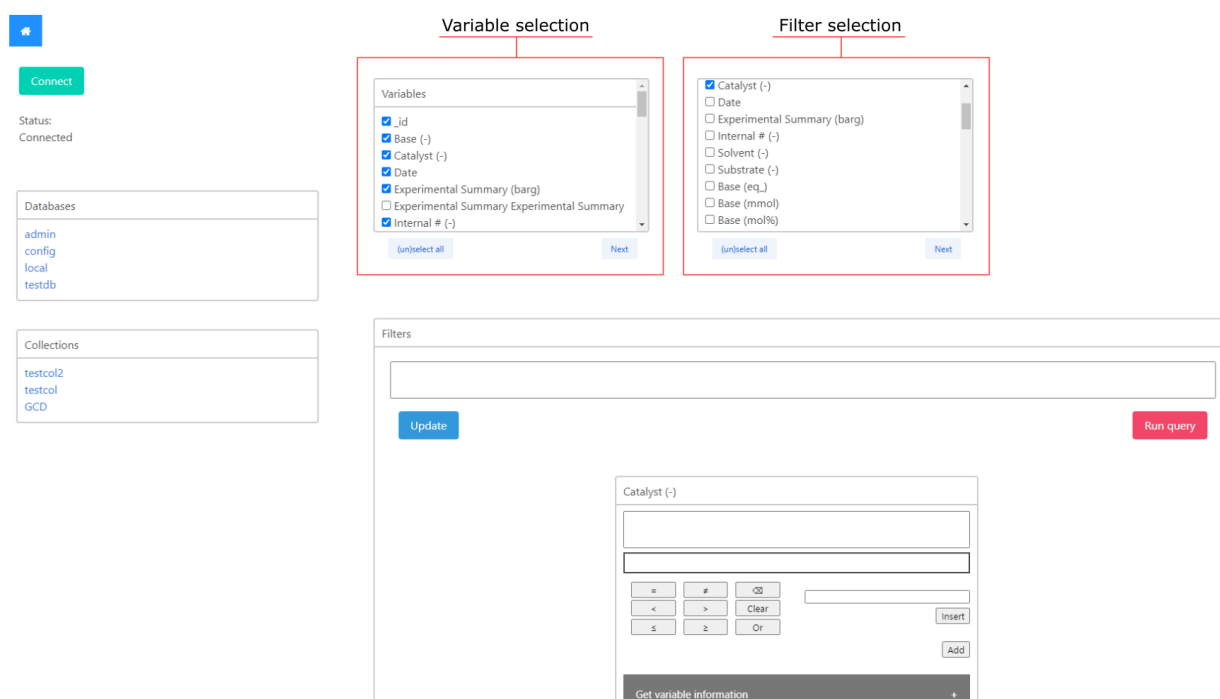


Figure 3.5: Platform page 3: Filter screen

For each selected filter variable a construction box is loaded. Figure 3.6 shows the construction box in the case of applying a single filter. Additionally, information should be provided about the stored variable values for the researcher to know what kind of conditions can be applied to a certain variable. For example, if the user would like to return experimental data with the highest yield from the database,

the user should know what the highest yield is in the stored experimental data. This information is provided to the user and the variable's range and minimum/maximum values are displayed at the bottom of the construction box. In the case of discrete variables all unique options are displayed.

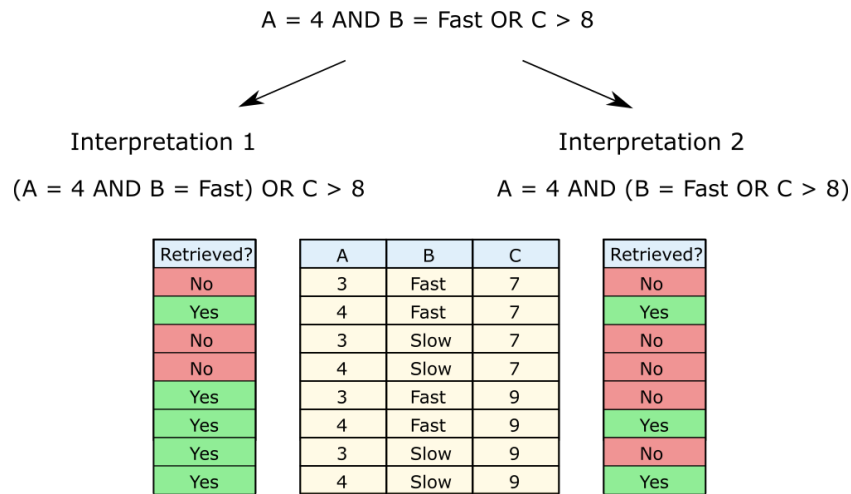
The screenshot shows a web interface for constructing filters. At the top, a text area displays a JSON query: 

```
{
  "Sand": [
    {
      "Date": {
        "Sgt": "2020-10-14T00:00:00.000Z"
      }
    }
  ]
}
```

. Below the text area are two buttons: a blue "Update" button and a red "Run query" button. In the center, a "Date" filter interface is shown. It has two input fields, both containing "> 2020-10-14". Below the input fields are several comparison operators: "=", "<=", ">=", "<", ">", "Clear", "≤", "≥", and "Or". To the right of these operators is a text input field containing "2020-10-14" and an "Insert" button. Below the operators is an "Add" button. At the bottom of the date filter interface, there is a section titled "Get variable information" which provides details about the "Date" variable: "The Date format is: YYYY-MM-DD e.g. 1997-12-15", "Earliest Date: Mon Apr 30 2018 02:00:00 GMT+0200 (Midden-Europese zomertijd)", and "Latest Date: Wed Jan 15 2020 01:00:00 GMT+0100 (Midden-Europese standaardtijd)".

**Figure 3.6:** Augmented view of the filter construction box

After specifying the desired conditions, the constructed database query will be shown at the top of the filter box. The user does not have to use this query to access the database, and it is instead provided to sanity check the system. This is important because there are specific queries that the current design is not capable of constructing. These complications are due to the ambiguous interpretation of combined conditions. To clarify, figure 3.7 shows an example of an ambiguous query, where two different sets of data are retrieved based on the query interpretation. The current design of the filter page cannot distinguish between these two interpretations and will consistently choose interpretation 1. This is not a problem for most queries and otherwise the the proper interpretation can be manually inserted into the query box.

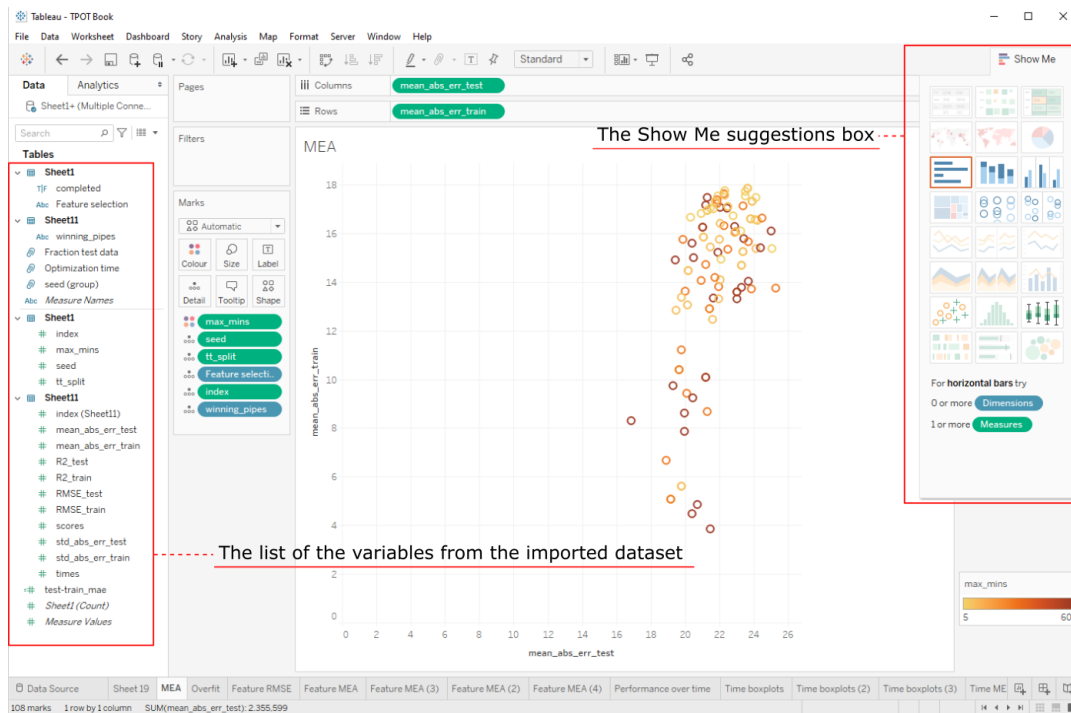


**Figure 3.7:** Example of an ambiguous query that leads to two possible interpretations

After construction of the query, MongoDB will return the matching data in an unstructured format. Finally, this data can be stored locally by the researcher in a data format of choice, for example as a comma separated values (CSV) file. This data could subsequently be used in one of the next steps of the platform.

### 3.4. DATA VISUALIZATION

Data visualization is a critical step in the process of constructing effective ML models, and is responsible for assessing the quality of the data. A first attempt at including visualization methods in a platform has shown the impracticality of manual integration.<sup>26</sup> More specifically, every type of visualization would need a specific piece of code to be hard-coded into the platform. Since useful graphs in data visualization can range from simple scatter plots to sophisticated multidimensional network diagrams, hard-coding is inefficient. Instead our platform opts to include data visualization with an external program made by Tableau. The intuitive design and convenient drag-and-drop functionality made Tableau a suitable choice for providing the visualization methods (an impression is provided in figure 3.8).



**Figure 3.8:** An impression of Tableau Desktop's design that is used in the platform

By including Tableau as the means for visualization, the user is required to install Tableau Desktop on its local machine. This software is free of charge for academic researchers.<sup>28</sup> After installation, the user can start Tableau from the platform's home screen. The data connections to Tableau are established at startup of the application. Tableau will automatically evaluate the imported data, and infer the data format and type. If necessary the data can also be modified, for example a newly calculated variable can be appended. The variables inside the dataset can be dragged onto the visualization canvas where they will be visualized (figure 3.8). The "Show Me" tool expedites the data visualization process by automatically suggesting common charts. Lastly, the completed visualization can be saved as an image or as an interactive Tableau workbook. To summarize, Tableau was integrated into the platform to provide data visualization functionality, and allows the researcher to efficiently explore and understand the data.

### 3.5. MODELLING

Once the data is prepared the process of model construction begins. The process of constructing the best ML model is often iterative and can even require revisiting a previous step. Various software frameworks and tools have been developed to assist the implementation of these steps. One of the most widely known ML frameworks is Scikit-Learn, which is an open-source Python package that facilitates the implementation of a wide range of ML algorithms on top of the Python programming

language. Since its first release in 2007 Scikit-Learn has been adopted by many companies, such as Evernote, J.P. Morgan, and Spotify.<sup>29</sup> In this work Scikit-Learn's exhaustive ML toolkit is applied to facilitate integration of an AutoML system.

### 3.5.1. AUTOML

Despite the easy access to ML algorithms, most steps in the ML workflow still require considerable ML expertise. One needs to consider the dimensions and quality of the data, the available training time, and the required interpretability of the model, to make the appropriate choice between the large number of available ML algorithms. Our analytics solution implements the concept of Automated Machine Learning (AutoML) through a virtual Python environment to sidestep the requirement of ML proficiency. As the name suggests, AutoML aims to automate the process of applying ML and thereby enable domain experts to automatically build ML applications without the requirement for extensive statistical and ML knowledge.<sup>30</sup> This reduction in user involvements is at the expense of the increase in computing costs that comes forth of AutoML's iterative approach in finding the best ML model structure. Currently, AutoML is still an active area of research, where the amount of expert knowledge is significantly reduced but not completely avoided. This knowledge is often required for initiating the AutoML process, where choices have to be made regarding the desired degree of interpretability of the ML model, as well as the fraction of data that should be used to validate the model's performance. Overall, the decisions in the AutoML process are more high-level, and require considerably less ML proficiency. Our platform integrates AutoML through the Python library TPOT.

TPOT (Tree-based Pipeline Optimization Tool) is built on top of the Scikit-Learn's package and systematically searches for a combination of ML modeling operations that maximizes the predictive performance of the final model. These operations involve data transformations, feature selection, algorithm selection, and hyperparameter tuning. A specific combination of these operations is referred to as a pipeline, two example pipelines are provided in figure 3.9. TPOT uses Genetic Programming to efficiently search through the large number of possible combinations for the best performing pipelines. Genetic Programming is a stochastic optimization algorithm inspired by the fundamental mechanisms of biological evolution.



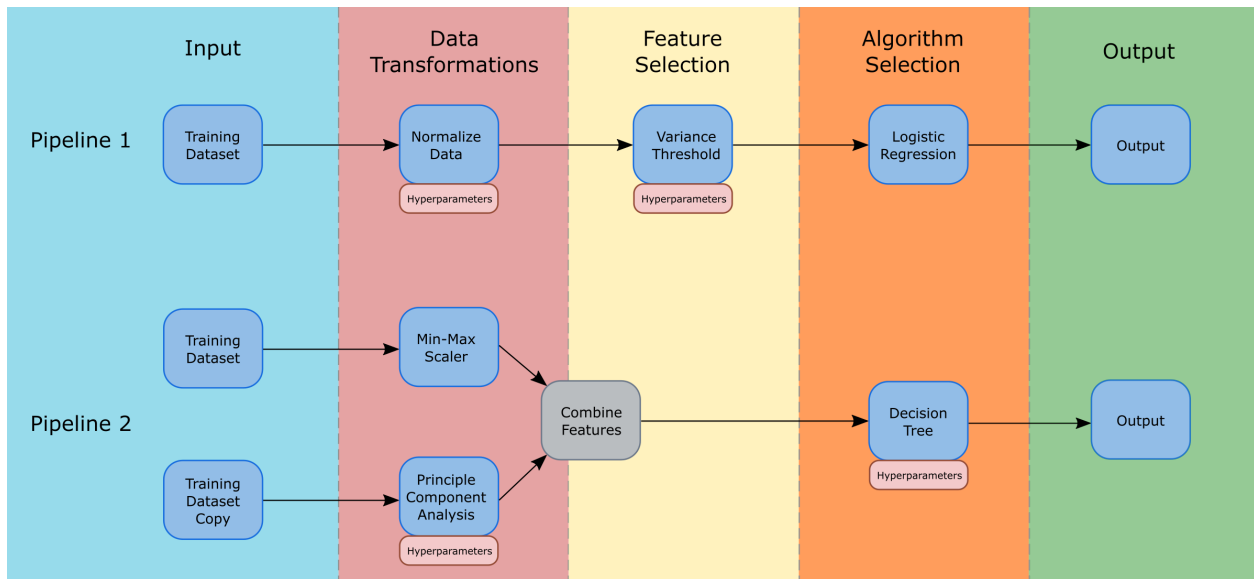


Figure 3.9: Two examples of ML pipelines that could result from TPOT's AutoML process

The optimization method starts by generating a large sample of pipelines, known as the population. TPOT evaluates the performance of these pipelines and creates a next generation population. The worst performing pipelines are excluded from this new generation while multiple variations of the best performing pipelines are included. These small variations, known as mutations, could involve using a different set of hyperparameters or adding a preprocessing step to the pipeline. These stochastic variations can positively or negatively affect the performance of the pipelines, and therefore allow TPOT to explore new pipelines that were never previously considered.<sup>31</sup> This process is repeated for a set number of iterations, known as generations, after which the best performing pipeline is chosen as the final ML model.

The process of AutoML is usually computationally intensive as it involves repeatedly training and evaluating ML pipelines. Our platform uses a server to provide remote computing power, and alleviates the user's local machine from these intensive computations. More specifically, the server uses JupyterHub which allows users to interact with a computing environment through a webpage. Although in this work JupyterHub is used to host a Python environment, the JupyterHub can be extended to also provide virtual environments for many other programming languages, including Julia, R, Matlab, and Scala.<sup>32</sup> The benefit of indirectly including a Python environment into the platform is also apparent when considering the future directions of the platform: newly constructed tools in Python or other programming languages can also be included into the platform through the JupyterHub. Normally the JupyterHub requires manual input through the interface of a webpage to be functional. Our platform overcomes this requirement through the web-automation tool, Puppeteer. Web-automation is the process of automatically performing preprogrammed operations. The most frequently encountered use case for web-automation is the development of bots, but in our case it

is used to connect the platform's input to the JupyterHub. The Puppeteer processes are initiated by the platform, and operate through headless instances of Google Chrome. These Chrome instances are used to access the JupyterHub and start Python processes.

The ML functionality is integrated into the final two platform pages. The first page (figure 3.10) is referred to as the training page and is designed to initiate the TPOT's AutoML process on the server's JupyterHub. The second page (figure 3.11) is referred to as the application page and is designed to apply and to provide insights of the constructed predictive models. The training page displays the necessary inputs that TPOT requires. These inputs have default values and generally do not have to be altered, instead the presented parameters serve as an option for the user to constrain TPOT's extensive search and, for example, limit the maximum training time or to reduce complexity of the final model. The latter is especially beneficial to increase the model's interpretability, which is commonly studied through feature importance.

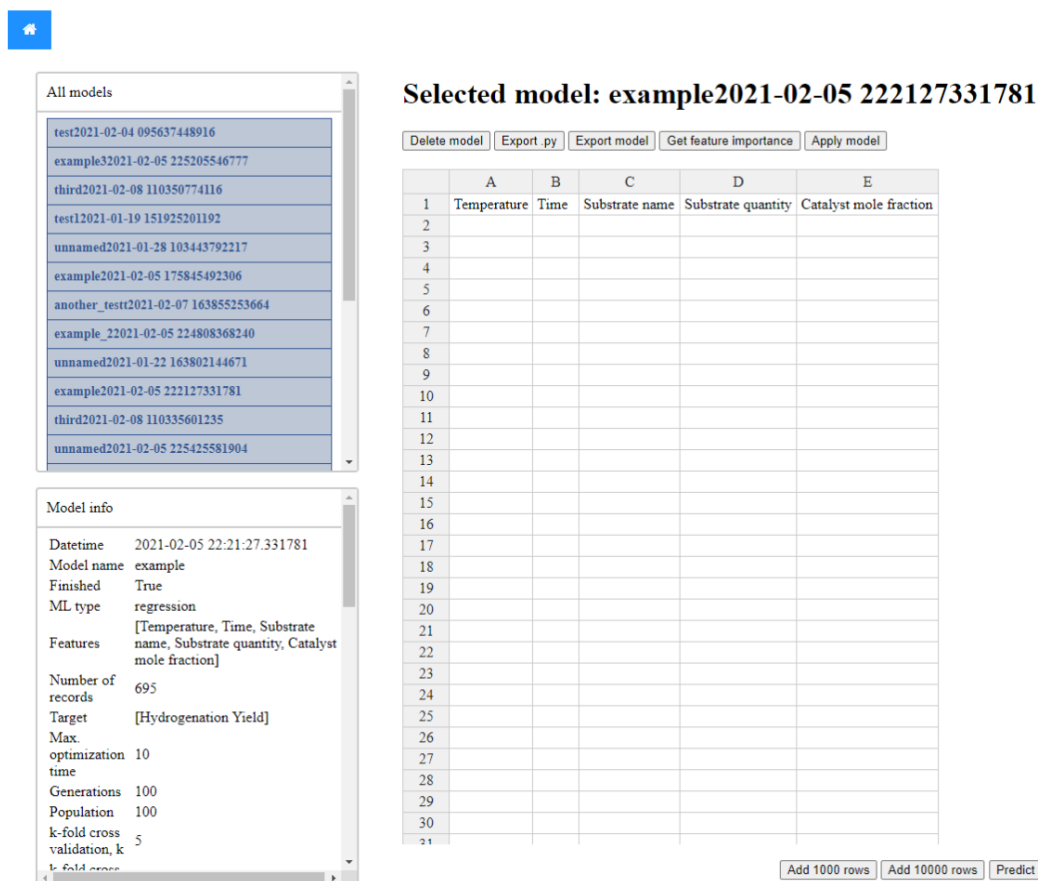
The screenshot displays the TPOT training interface. On the left, there is a vertical list of input fields: 'Model Name' (text input, 'unnamed'), 'Model Type' (dropdown menu, 'Regression'), 'Population Size' (text input, '100'), 'Generation Size' (text input, '100'), 'Maximum Time (minutes)' (text input, '-'), 'Cross Validation K' (text input, '5'), 'Cross Validation Repeats' (text input, '5'), 'Seed' (text input, '1234'), 'Test Split Fraction' (text input, '0.1'), and 'Template' (dropdown menu, 'Stacking'). Below these is a 'Start Training' button. In the center, a 'Features' panel contains a list of features with checkboxes: Reactor Type (checked), Reactor Volume (unchecked), Temperature (checked), Pressure H2 (checked), Time (checked), Substrate name (checked), Substrate SMILES (unchecked), Solvent density (unchecked), Solvent volume (unchecked), Solvent quantity (checked), Solvent mole fraction (unchecked), Solvent.1 (unchecked), Loading procedure (unchecked), Loading sequence (unchecked), Additives (checked), Quantity (unchecked), Additive concentration (checked), Reported yield type (unchecked), Acyl Alcohol Yield (unchecked), Transester Yield (unchecked), tBu Ester Yield (unchecked), Ether Yield (unchecked), Hydrogenation Yield (unchecked), S/C (unchecked), TON (unchecked), TOF (unchecked), Reference DOI (unchecked), and Coordination\_center (checked). At the bottom of this panel is a '(un)select all' button. On the right, a 'Target' dropdown menu is open, showing a list of target variables: Acyl Alcohol Yield, Transester Yield, tBu Ester Yield, Ether Yield, Hydrogenation Yield (highlighted in blue), S/C, TON, TOF, and Reference DOI.

Figure 3.10: Platform page 4: Training screen

Feature importance refers to techniques that assign a score to the input features of a predictive ML model based on how useful they are for predicting the target variable. For most pipelines this method could lead to insight into the data and the model's logic, but its effectiveness is reduced for long pipelines with many preprocessing and data transformation steps. The reason is that these steps result in a set of features that is completely different from the features in the provided data set, this could for example occur through principal component analysis (PCA). As a result, interpreting the importance of the original features is virtually impossible because the feature importance scores consider the transformed features. In order to provide the option of maintaining interpretability, the training page

offers the option to construct simpler pipelines without preprocessing or data transformation steps.

Upon opening the application page, the top-left box displays all of the initiated AutoML processes. More specifically, the box contains finished ML models as well as models where the AutoML process has yet to finish. When selecting one of the models, the box on the bottom-left of the page will load information about the model. These include the settings for TPOT as well as performance metrics of the trained ML model.



The screenshot displays the application interface. On the left, there is a list of models under the heading "All models". The selected model is "example2021-02-05 222127331781". Below the list is a "Model info" section with the following details:

- Datetime: 2021-02-05 22:21:27.331781
- Model name: example
- Finished: True
- ML type: regression
- Features: [Temperature, Time, Substrate name, Substrate quantity, Catalyst mole fraction]
- Number of records: 695
- Target: [Hydrogenation Yield]
- Max. optimization time: 10
- Generations: 100
- Population: 100
- k-fold cross validation, k: 5

On the right, the "Selected model: example2021-02-05 222127331781" is displayed. Below the title are buttons for "Delete model", "Export py", "Export model", "Get feature importance", and "Apply model". A spreadsheet is shown with the following columns: A (Temperature), B (Time), C (Substrate name), D (Substrate quantity), and E (Catalyst mole fraction). The spreadsheet has 31 rows, with the first row containing the column headers and the rest being empty. At the bottom right of the spreadsheet are buttons for "Add 1000 rows", "Add 10000 rows", and "Predict".

Figure 3.11: Platform page 5: Application screen

In addition, the training page provides several functions that are implemented through the buttons at the top of the page and the central spreadsheet. Two of these buttons relate to the page's export functionality, where the user can distinguish between exporting a Python script with the code for the model's architecture, and a model file that was fully trained on the provided dataset. This page also provides the option to inspect the model's feature importance. As shown in figure 3.12 each of the features are presented in the spreadsheet with their relative importance. The main function of this page is to apply trained ML models. The features required for ML predictions are loaded inside of the spreadsheet on activating the apply function. Subsequently, the new data input can be deposited inside of this spreadsheet. Using the ML models' predictive power inside of the platform

is advantageous because the required data format for the model is clearly presented, which allows the user to effortlessly make predictions. Finally, the ML model's predictions for the combination of features will be added in an additional column for the target variable. These predictions conclude the ML process.

**Selected model: name\_of\_model**

	A	B	C
1	Feature	Weight	Stdev
2	Catalyst quantity	0.7041264739338084	0.1633607799190641
3	Solvent mole fraction	0.12732539227672893	0.15292339425217044
4	Time	0.10200640050832492	0.09082465933686198
5	Temperature	0.06654173328113776	0.09464305698704124

**Figure 3.12:** An augmented view displaying the results of the platform's feature importance functionality

In summary, a data management and analytics platform was successfully developed. The resulting desktop application involves an infrastructure of both hardware and software technologies. Through this platform catalysis researchers can store their data in a database, and use their data to build predictive ML models. Conventionally, the construction of these models is demanding in terms of ML proficiency. Our platform largely eliminates this requirement, thereby making ML more accessible to the researcher. With the prospect of increasingly sized catalysis databases, the potential of data-driven techniques such as ML is promising. In the remainder of this work, this potential will be explored by applying the platform in a case study.

# 4

## CASE STUDY

In this chapter the functionality of the developed platform is validated by means of a case study. At the moment of writing large structured databases are not readily available in the field of catalysis. As a result, we are unable to demonstrate the platform's capabilities on an existing dataset. Instead, an artificial dataset was constructed to demonstrate the potential of the developed method. In this dataset the underlying principles and mechanisms describing the data are manually determined, and designed to mimic catalytic data. This allows us to emulate the future scenario when sufficiently sized databases are available, as well as to gain insights into the requirements for future catalytic databases. The added benefits of an artificial dataset is that the true relations between the variable are known which makes the evaluating process more insightful.

In catalysis informatics, there are a number of available experimental datasets that have previously been used to demonstrate ML and other statistical techniques on catalysis data. Among these datasets, the most sizeable databases consist of several thousand data points.<sup>33–35</sup> Unfortunately, these datasets are compiled from publications by different groups, which risks introducing bias into the dataset. Additionally, the datasets report global properties, such as selectivities, yields and conversions, as the measure for catalytic activity. These properties are often based on the final concentrations of the catalytic mixture, and fail to reflect the time-dependent kinetic behavior. The total catalytic performance is insufficiently evaluated by these thermodynamic measures because catalytic reactions are often controlled kinetically, rather than thermodynamically. Therefore, we expect different datasets to be constructed in the future, where kinetic information is included in the datasets, that can be used to train ML models. To clarify the importance of kinetic properties in catalyst assessment, figure 4.1 shows the yield profiles of two different catalysts. Although the first catalyst has a slightly higher final yield, the second catalyst reaches the 80th yield percentage much faster. Measuring catalytic activity through the yield indicates a similar catalytic performance between the two catalysts. However, when considering the transient kinetic behaviour it becomes clear that the second catalyst exhibits preferable catalytic activity as it has a much higher reaction rate. The importance of kinetic data has led us

to propose that it will be included in future catalytic datasets, and it will therefore be included in the artificial dataset as well. More specifically, the artificial dataset includes the kinetic performance of a large number of catalysts along with chemical descriptors for these catalysts, and aims to demonstrate the potential of a data-driven workflow, as well as to provide insights about the future directions for catalytic datasets.

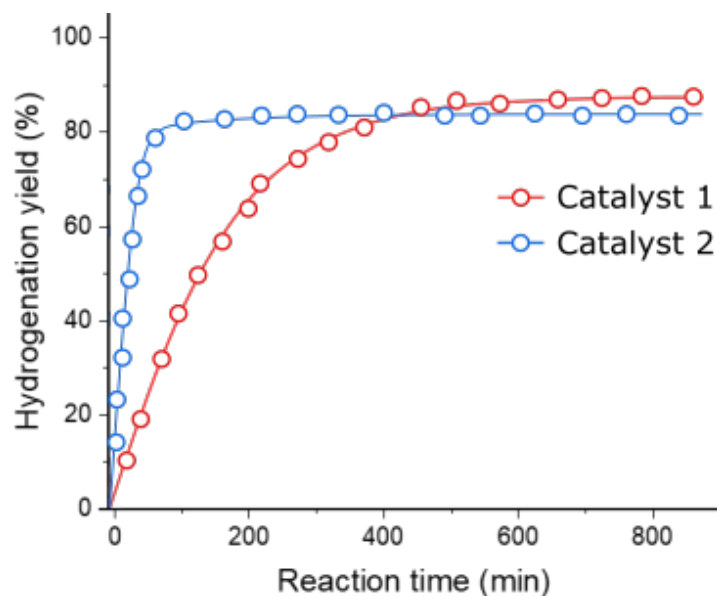
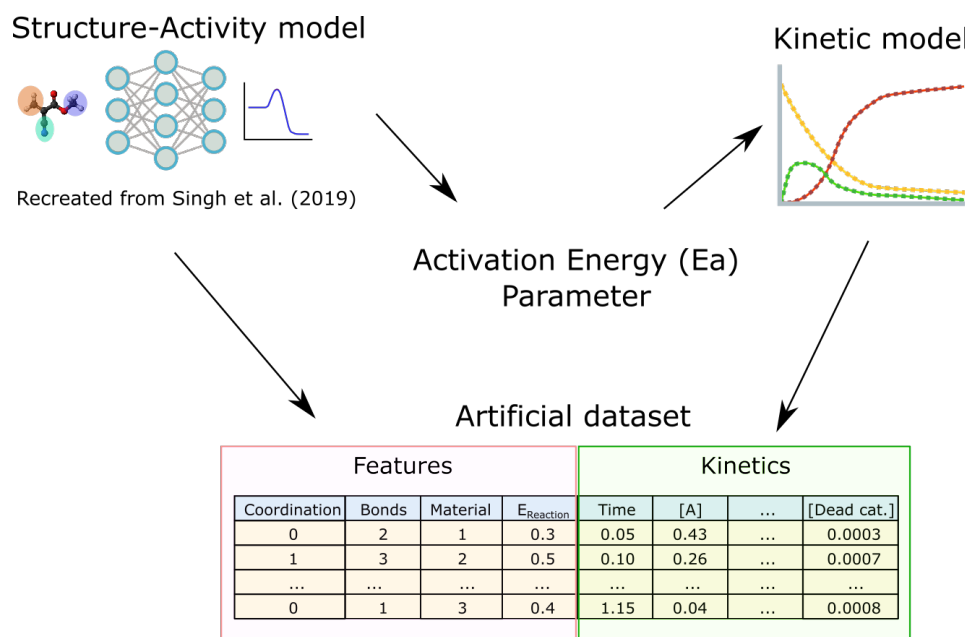


Figure 4.1: An example of product concentration profiles for two different catalyst

## 4.1. DATASET CONSTRUCTION

The artificial dataset is constructed by combining two models: a structure-activity model obtained from literature, and a kinetic model. These models serve as the underlying principles of the data, and are used to generate structural and kinetic data for hypothetical catalysts. More specifically, the molecular descriptors of a catalyst are related to an activation energy of a specific target reaction through the structure-activity model, which is subsequently used in the kinetic model to generate the transient concentration data. A global overview of the construction process is shown in figure 4.2. The activation energy, which will be introduced as the  $E_a$  parameter, is a central parameter that forms the link between the structure-activity model and the kinetic model. In this dataset, the catalysts have different values for their set of molecular features, which corresponds to a unique value for the  $E_a$  parameter, and thus lead to different kinetics for the catalysts under the same reaction conditions. Consequently, the number of catalysts in the dataset is simple to control because our combination of models only requires a value for the  $E_a$  parameter to generate the structural and kinetic data of a new catalyst. It should be noted that in actuality the structure-activity model can only be used to generate an  $E_a$  parameter from the molecular descriptors, and not vice versa. Nevertheless, specific values for the  $E_a$

parameter can still be obtained by providing the structure-activity model with a large random sample of descriptor values, and subsequently searching through the resulting  $E_a$  parameters for the desired value.



**Figure 4.2:** Overview of the artificial database construction process

Through this method of construction, the information stored in an  $E_a$  parameter is conveyed to the molecular descriptors, which will be used as the features for the ML models. The reason for using the molecular descriptors as the features instead of the  $E_a$  parameter is that in catalysis research activation energies are very complex to measure or expensive to calculate. This is problematic for the construction of large databases, as well as for application of ML models that are trained on this data because obtaining the features for resulting ML predictions would require considerable effort. Including the  $E_a$  parameter in the artificial dataset would therefore obstruct it from being a representative dataset for a realistic catalytic dataset. In contrast, using molecular descriptors that correlate with the  $E_a$  parameter do not have this problem as long as they can be obtained with relative ease.

The following two sections will explain the workings of the kinetic and the structure-activity model, respectively.

#### 4.1.1. KINETIC MODEL

The kinetic models consists of a system of chemical reactions, which is used to generate the kinetic data for the artificial dataset. Although, the chosen chemical reactions are hypothetical and arbitrary examples, they should contain enough complexity to represents a realistic scenario within catalysis. Therefore, the governing chemical reaction pathway is chosen such that it maximizes the applicability

of this case study.

Reactions in catalysis are very diverse, and include for example both hydrogenation and cross-coupling reactions. This diverse nature of catalysis makes it virtually impossible for a single chemical reaction pathway to represent all of the reactions within catalysis. However, there are common factors between complex catalytic systems that are important to take into consideration, such as deactivation and side reactions. Our kinetic model includes catalyst deactivation and 7 chemical species to ensure sufficient complexity in this case study. The devised chemical reaction pathway is shown in figure 4.3. The desired reaction converts the substrate (species A) to the product (species B). In addition, a multi-step side reaction involves the conversion to an undesirable side product (species E) through 2 intermediates (species C & D). The remaining 2 species are the activated and deactivated form of the catalyst. In constructing the model that governs the kinetics of this pathway, the reaction steps are chosen to be irreversible first order and the reaction rate constants are chosen to obey Arrhenius equations. This leads to reaction rate expressions of the form shown in equation 4.1, where  $i$  and  $j$  respectively represents the product and reactant of a reaction. In this equation the temperature ( $T$ ) is constant at 298.15K and the  $C_{cat}$  term is only added if the reaction is catalyzed.

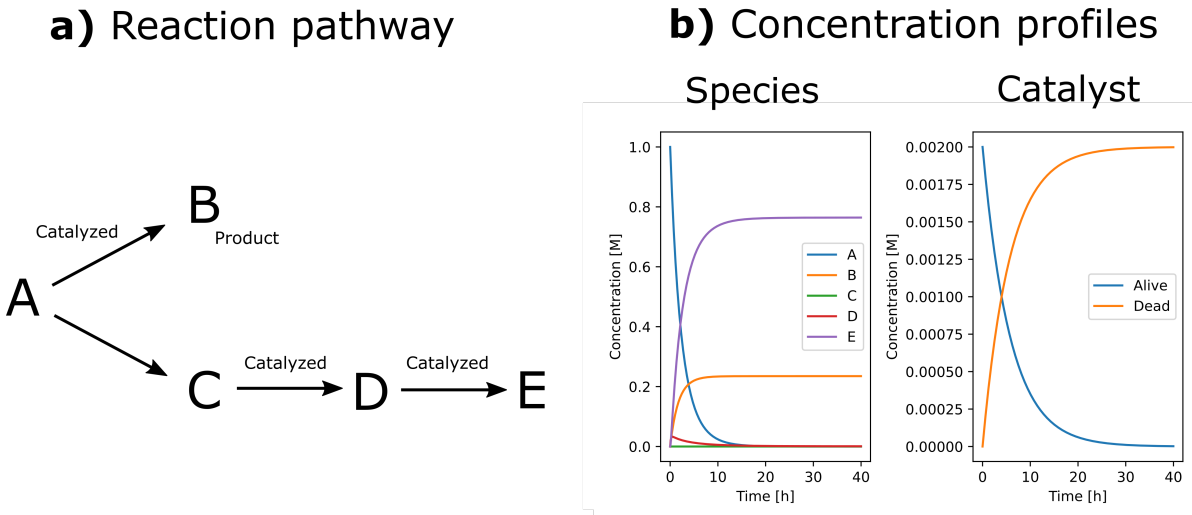


Figure 4.3: The reaction kinetics for the chemical pathway used in the case study

$$r_i = A_i e^{\frac{-Ea_i}{RT}} C_j C_{cat} \quad (4.1)$$

$$R_i = \frac{dC_i}{dt} = \sum v r \quad (4.2)$$

A coupled system of ordinary differential equations (ODEs) is produced by defining the production rate expressions as shown in equation 4.2. In this expression  $v$  is the stoichiometry of species  $i$  in the



reaction governed by the reaction rate expression  $r$ . Table 4.1 shows the values for the kinetic constants that were used in evaluating the obtained expressions. The activation energy parameter for the reaction that forms the main product is most influential on the resulting transient compositions of the reaction mixture. Throughout this work this activation energy parameter is referred to as the  $E_a$  parameter. The  $E_a$  parameter is varied between 50 and 70  $\text{kJmol}^{-1}$  to represent different catalysts. These specific numbers are chosen because outside of this range the concentration profiles are unaffected by this parameter. In this approach, every catalyst in the final dataset has a unique value for this parameter within the mentioned range. For each of the catalysts, the system of ODEs is numerically solved with a time resolution of 1200 point to obtain the concentration profiles (figure 4.3), which forms the first part of the artificial dataset. This extremely high time resolution was chosen as a precautionary measure to avoid the possible requirement of a larger time resolution at a later point in the thesis project.

**Table 4.1:** The constants used in the kinetic model

Reaction	Reactant	Product	Pre-exponential factor ( $s^{-1}$ )	Activation energy ( $\text{kJmol}^{-1}$ )	Initial reactant concentration ( $M$ )
1	A	B	$1 \cdot 10^{12}$	Between 50 and 70	1
2	A	C	$1 \cdot 10^{10}$	60	0
3	C	D	$1 \cdot 10^{16}$	20	0
4	D	E	$1 \cdot 10^8$	25	0
5	Activated cat.	Deactivated cat.	$1 \cdot 10^8$	50	$2 \cdot 10^{-3}$

#### 4.1.2. ACTIVATION ENERGY CORRELATION

As previously discussed, the information contained in the  $E_a$  parameter can be indirectly included in the catalytic dataset by including features that strongly correlate with the  $E_a$  parameter. For example, within heterogeneous catalysis it has been shown that there is a linear relationship between the activation energy and the enthalpy change of an elementary reaction, known as the Brønsted-Evans-Polanyi (BEP) relation.<sup>36</sup> A concrete formulation of the relation between these descriptors and the activation energy is necessary to include descriptors for the activation energy into the artificial dataset. Linking multiple descriptors to activation energy has currently not been achieved by theoretical models, however, efforts in ML have been able to develop models of the form shown in equation 4.3.<sup>37,38</sup>

$$E_a(\text{descriptor}_1, \dots, \text{descriptor}_n) \quad (4.3)$$

In this work we integrate the constructed ML model from Singh et al. (2019) as the underlying relation between descriptors and the  $E_a$  parameter for our artificial dataset. Their model was chosen because

it was the most accurate and the most recently published attempt at connecting molecular descriptors to activation energy. In their research, a ML model was trained on dissociation reaction data obtained from DFT calculations.<sup>38</sup> Their model predicts the activation energy of the reaction based on 4 features: (1) coordination of the surface, (2) the number of bonds broken between the initial and final state, (3) the identity of the surface atom involved in bond breaking, and (4) the reaction energy. In particular the reaction energy showed to be a powerful feature in prediction of activation energies. This result was to be expected because both the reaction energy and activation energy for simple reactions correlate with the d-band center of a metal catalyst.<sup>15</sup> Opposed to the activation energy, these features are cheap to determine, and thus suitable for the construction of large databases. In the case of the reaction energy DFT calculation are required, but these calculations are significantly cheaper than the calculations of transition states. In their results, they report a mean absolute error of roughly 19.3 kJ/mol which, relative to the average magnitude of their data points, correspond to an average relative error of 13.8%. By recreating their ML model we were able to connect the  $E_a$  parameter to the above mentioned features. This results in a final dataset with catalyst features that correlate with the activation energy and in turn with its kinetic performance.

In summary, the dataset construction process starts by generating a large sample of activation energies by providing the structure-activity model with random combinations of its features. The activation energies of interest, i.e. those between 50 and 70  $\text{kJmol}^{-1}$  (section 4.1.1), are filtered from the large sample. The combination of features that generated these filtered  $E_a$  parameters makes up the first half of the data. The second half is the transient composition data, which is generated by providing the kinetic model with the filtered  $E_a$  parameters. This process led to a final artificial dataset that includes features and kinetic data for 946 catalysts.

### 4.1.3. DATASET VARIATIONS

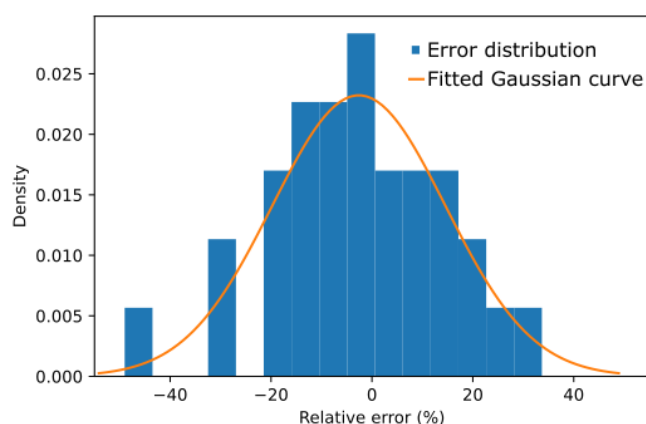
A large advantage of an artificially constructed database over one containing real catalytic data is the control over underlying mechanisms and database size. In this work we make use of this aspect by imposing variations in the dataset to measure the effect of data quality and quantity on the ML performance. More specifically, dataset variations were constructed where database size and descriptor strength were varied. By quantifying the effects of these variations we aim to reveal useful information regarding the necessary objectives for future construction of databases from real catalytic data.

The first variation aims to uncover the necessary database size required for ML techniques to lead to helpful insights. Determining the appropriate database size could give an indication on the required database size for future work in catalysis informatics. However, it has to be mentioned that the relevance of the findings will vary for each specific case in catalysis.

The size variations for our artificial database included 700, 500, 300, 100, and 50 catalyst next to the

original dataset with 946 catalysts. In constructing these dataset variations, the selected  $E_a$  parameters were chosen such that a uniform distribution was attained. This uniformity corresponds to having equal numbers of well and poor performing catalysts in the dataset, and is important to avoid bias in the data.

The second variation aims to uncover the influence of the descriptor strength. Essentially the descriptor strength is a measure of the theoretically achievable performance of a ML model given the set of features. For example, when given the objective of predicting the phase of water, a dataset containing only one feature with temperature data has a lower descriptor strength than a dataset containing features with temperature and pressure data. In the original artificial dataset, features are included that were directly used in the process of generating the transient composition data. This means that the best possible ML model for this dataset could in theory perfectly predict the composition data from the features without error. In reality obtaining a set of features that contains all the necessary information for flawless predictive capability is unlikely. Therefore, an error is manually induced to weaken the descriptor strength, and mimic scenarios where the features only partially possess the information necessary for flawless predictions. More specifically, for each of the catalysts in the dataset an error added to their  $E_a$  parameter, thereby weakening the correlation between the features and the  $E_a$  parameter. The first variation of descriptor strength is based on the previously discussed publication by Singh et al. (2019). Their results showed an average relative error of 13.8%, where the errors roughly follow a normal distribution (figure 4.4). Therefore, the manually induced errors are also chosen to follow a normal distribution. In addition, two more dataset variations with an induced error of 1% and 5% relative to the average value of our data were constructed. These variations represent more optimistic scenarios regarding the descriptor strength that can possibly be achieved in future datasets.



**Figure 4.4:** Error distribution of the recreated ML structure-activity model from Singh et al. (2019)

The final variation to the original artificial dataset concerns the time resolution for the concentration points, which is set to 1200 points. This is much higher than the time resolution that can currently be

achieved in catalysis experiments, but allows for a more accurate measure of the performance of the ML model. A model is trained on a smaller time resolution of only 25 concentration points, to verify that the unrealistically high time resolution does not affect the model's performance

Finally, each of the ML models were obtained through TPOT's AutoML process that is integrated into the platform, and the training processes were initiated with a test split size of 10%, a population size ranging from 25 to 50, and a number of generations ranging from 10 to 30. Afterwards, these models are compared regarding their predictive capability.

#### 4.1.4. EVALUATION METHOD

Before we could look at these numbers we had to find a way to properly assess model performance. This work includes two evaluation methods, a standard method of evaluation and a method where the model is applied to predict catalytic performance. The standard method of evaluating ML model's performances is to withhold a random selection of data points from the ML algorithm during the training process. This selection of data points is used afterwards as a test set to measure the accuracy of the ML model on unseen data. In context of this case study, the test split method is able to quantify the average accuracy of predicting the time-dependent concentration points, and able to detect patterns across the dataset variations. The drawback of a random selection of data points is that for every catalyst the model has likely "seen" some of its concentration points during the training process. A hypothetical scenario is illustrated in figure 4.5, where performance of a trained ML model is tested by the purple data points. In reality a ML model of this kind will be applied in catalyst discovery to data that it has never seen before, which makes the standard test method unrepresentative of the real applications of the model. Therefore, a second method of evaluation was used to emulate a realistic scenario for ML model evaluation. We will refer to this method as the applicative method of evaluation. In this method the ML model is presented with 20 catalysts of mixed catalytic performance. The objective of the model is to correctly infer the most promising catalysts through its predictions for the transient composition data. Opposed to the standard method of evaluation the ML model has not experienced any of this data during the training process, which makes the applicative test method representative of the real applications of the model.

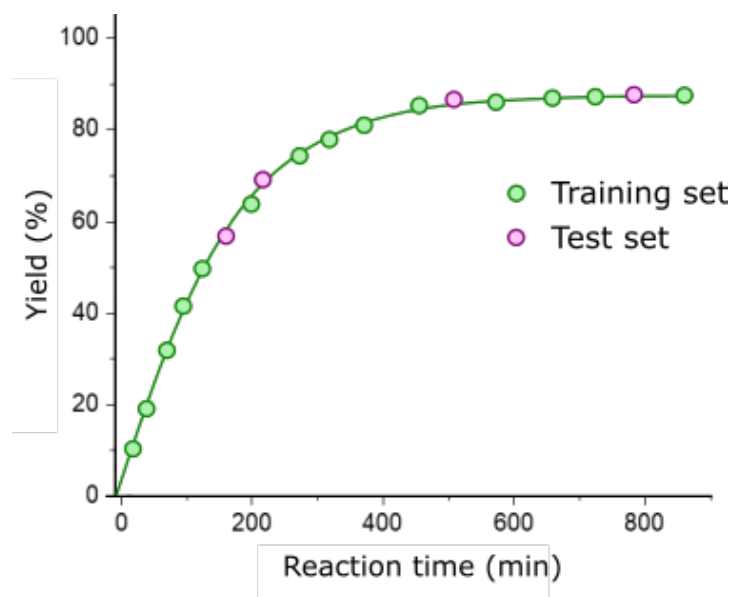


Figure 4.5: Example distribution of training and test set data points for a concentration profile

The applicative method of evaluation assesses catalytic performance through two parameters, which are shown in figure 4.6. The first parameter is the final concentration (FC) and the second parameter is the initial reaction rate (IRR). Both of these parameters can be decisive in judging a catalyst's performance. The IRR captures the transient performance of the catalyst, while the FC contains information about the thermodynamic performance. During the applicative evaluation method, these parameters are inferred from the predicted transient concentration data.

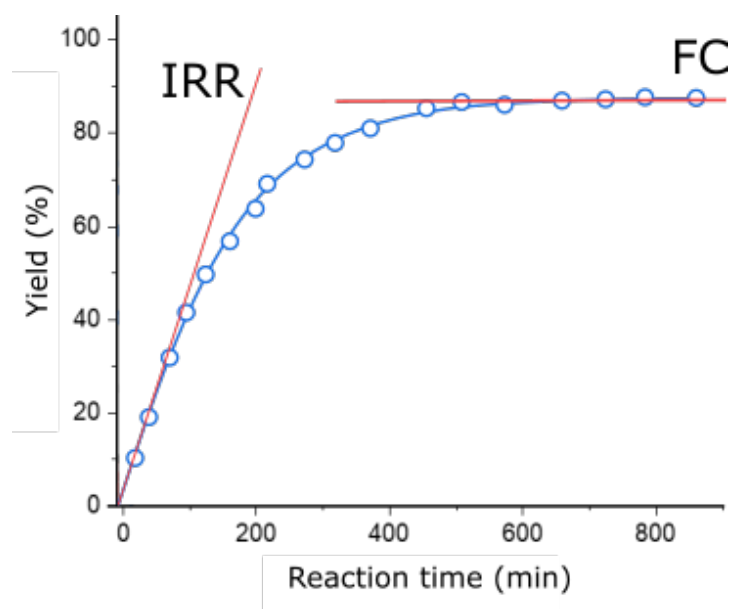


Figure 4.6: Exemplary graph with the two parameters for assessing catalytic performance: final concentration (FC) and initial reaction rate (IRR)

With a combination of the two methods of evaluation standard ML performing metrics are acquired as well as more insightful results that relate to the application of the ML models in catalyst discovery. In addition, the effectiveness of the standard method of evaluation can be compared to the applicative method in terms of suitability for assessing ML models that are to be used in catalyst discovery. During this evaluation process the predictive performance of all ML models is quantified through metrics.

Within ML many different metrics are used to evaluate model performance. The selection process that led to choosing the metrics for this work were mostly influenced by the interpretability of the results. In this work the mean absolute error (MAE) is chosen, which is shown in equation 4.4 where  $y_i$  and  $\hat{y}_i$  are the true and predicted value respectively. Opposed to other metrics the changes in the MAE scale linearly, and its units match that of the target variable. These aspects make the MAE an intuitive metric. Instead of an absolute error representation, the error metric will be presented in a relative form. This makes the results more generally applicable to similar future research, instead of just this particular case study. The relative error between groups of data points or vectors is shown in figure 4.5, where the enclosing double lines denote the "norm" of the enclosed vectors. The norm of a vector is a quantity that describes the vector's magnitude. There are many different options for calculating a norm, but in this work, only the Manhattan norm is used. This norm, which is also known as the L1 norm, is calculated by summing the absolute values of the vector (equation 4.6). The main reason for using the Manhattan norm in this work is the resulting interpretability for the relative error. In our specific case, where the data points are positive concentration points, the combination of the Manhattan norm and the relative error can be rearranged to equal the ratio of the MAE to the average value in the vector (equation 4.7). More specifically, when applied to the concentration points, the relative error signifies the MAE of the concentration predictions as a percentage of the average true values of the concentration.

$$\text{MAE} \equiv \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{n} \quad (4.4)$$

$$\text{Relative error} \equiv \frac{\|\mathbf{y} - \hat{\mathbf{y}}\|}{\|\mathbf{y}\|} \cdot 100\% \quad (4.5)$$

$$\text{Manhattan norm} \equiv \|\mathbf{y}\|_1 = \sum_{i=1}^n |y_i| \quad (4.6)$$

$$\frac{\text{Rel. error}}{100\%} = \frac{\|\mathbf{y} - \hat{\mathbf{y}}\|_1}{\|\mathbf{y}\|_1} = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|}{\sum_{i=1}^n |y_i|} = \frac{\sum_{i=1}^n |y_i - \hat{y}_i|/n}{\sum_{i=1}^n |y_i|/n} = \frac{\text{MAE}}{\text{Average } y} \quad (4.7)$$

## 4.2. RESULTS & DISCUSSION

In this section we will first discuss the results from the standard ML evaluation technique and thereafter the results from the applicative method of evaluation. The results regarding the dependence of the time resolution are not included but it was confirmed that the time resolution does not affect the ML models' performance on predicting the transient composition data within the investigated boundaries.

First the results of the database size variations are examined. Figure 4.7 shows the relative error in prediction of the test set with varying database size. In this figure no apparent trend is visible regarding the influence of the database size on the performance of the ML models. Moreover, the relative errors are small, at less than a quarter of a percent. A possible explanation for this result is that datasets that contain only 50 catalysts are already sufficiently sized for ML methods to be maximally effective, and that the deviations across the different sizes are due to the stochastic nature of ML and the AutoML process. However, it is known from the fields of statistics and data science that the amount of data is in some form proportional to predictive power. These results do not reflect this principle which makes them counter-intuitive.

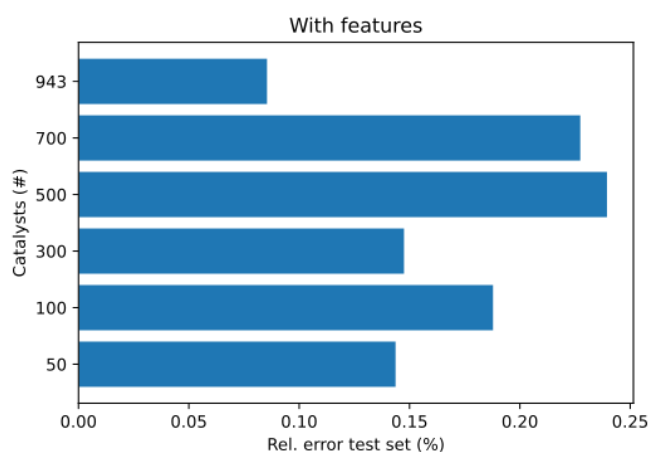
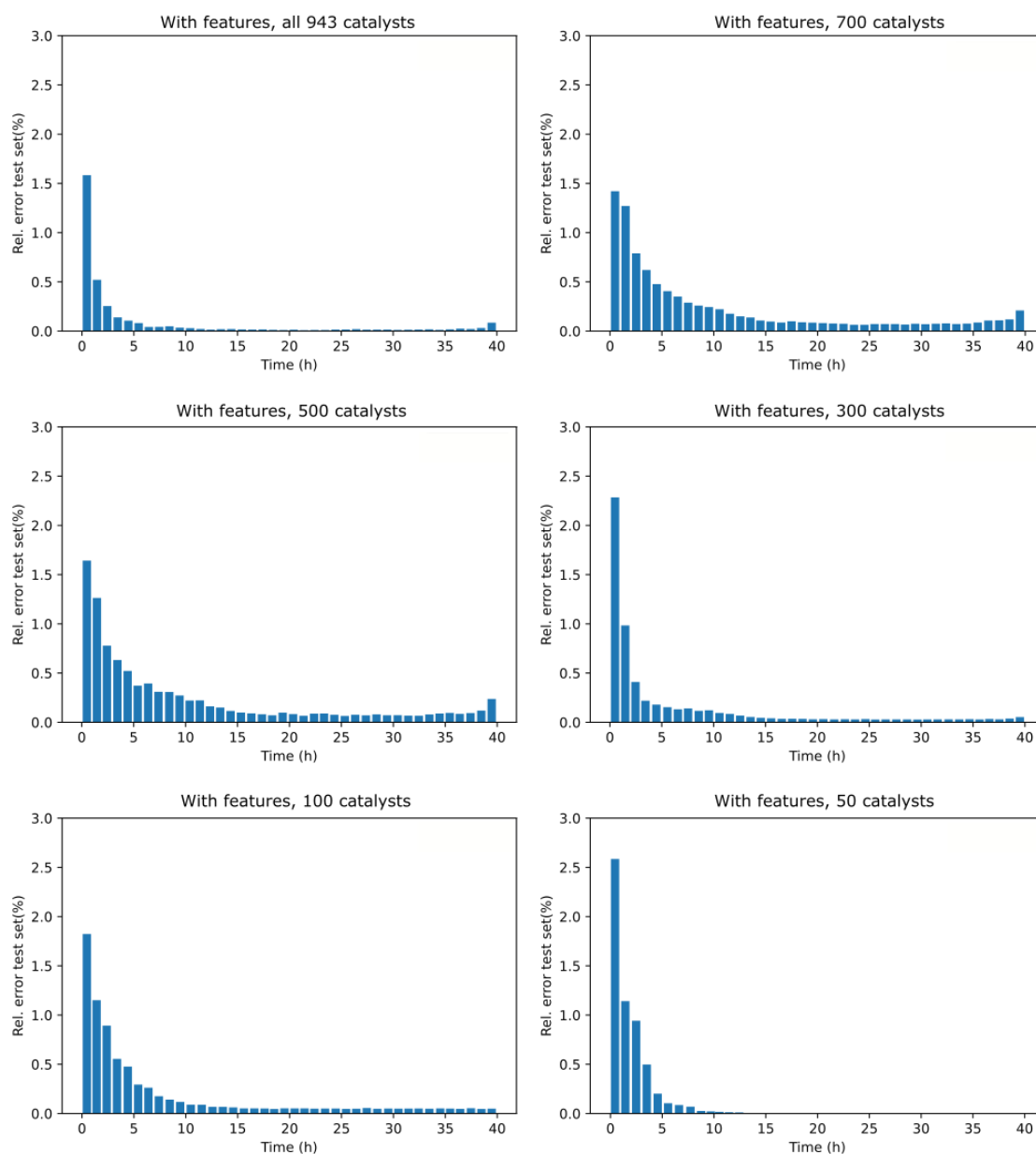


Figure 4.7: The ML models' test set performances with varying database sizes

It is also important to understand what the results in figure 4.7 fail to reveal, which is the distribution of the relative errors across the extent of the reaction. The rates and changes in the product concentration are highest at the start of the reaction, which makes the concentrations near the start the most challenging to predict. This effect is clearly illustrated in figure 4.8, where the relative error is grouped in 40 time intervals. In this figure the expected pattern is displayed and larger datasets result in better predictive performance. Additionally, it becomes evident why the relative error over the entire test set fails to present a similar pattern. Around roughly 15 hours the equilibrium concentration is reached, after which each of the models is able to predict the concentration with small error. Although all of

these small errors seem insignificant, the large number of these equilibrium concentration points has a significant influence on the average over the entire time domain. In effect prolonging the length of an experiment that has already reached equilibrium will result in a relative error metric that is unreasonably affected by small errors. We define this phenomenon where the average error is highly influenced by small insignificant errors as small-error stacking.



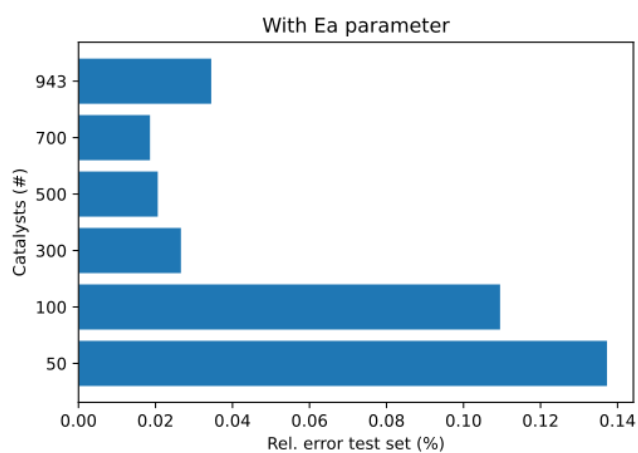
**Figure 4.8:** A view of the time dependence of the test set performance with varying database sizes

In addition, ML models were also trained on data where the  $E_a$  parameter is provided as a feature in the dataset (figure 4.9). These models aim to reveal to what extent the error can be attributed to the complexity of the underlying relation. Throughout all of the figures, the models trained on the

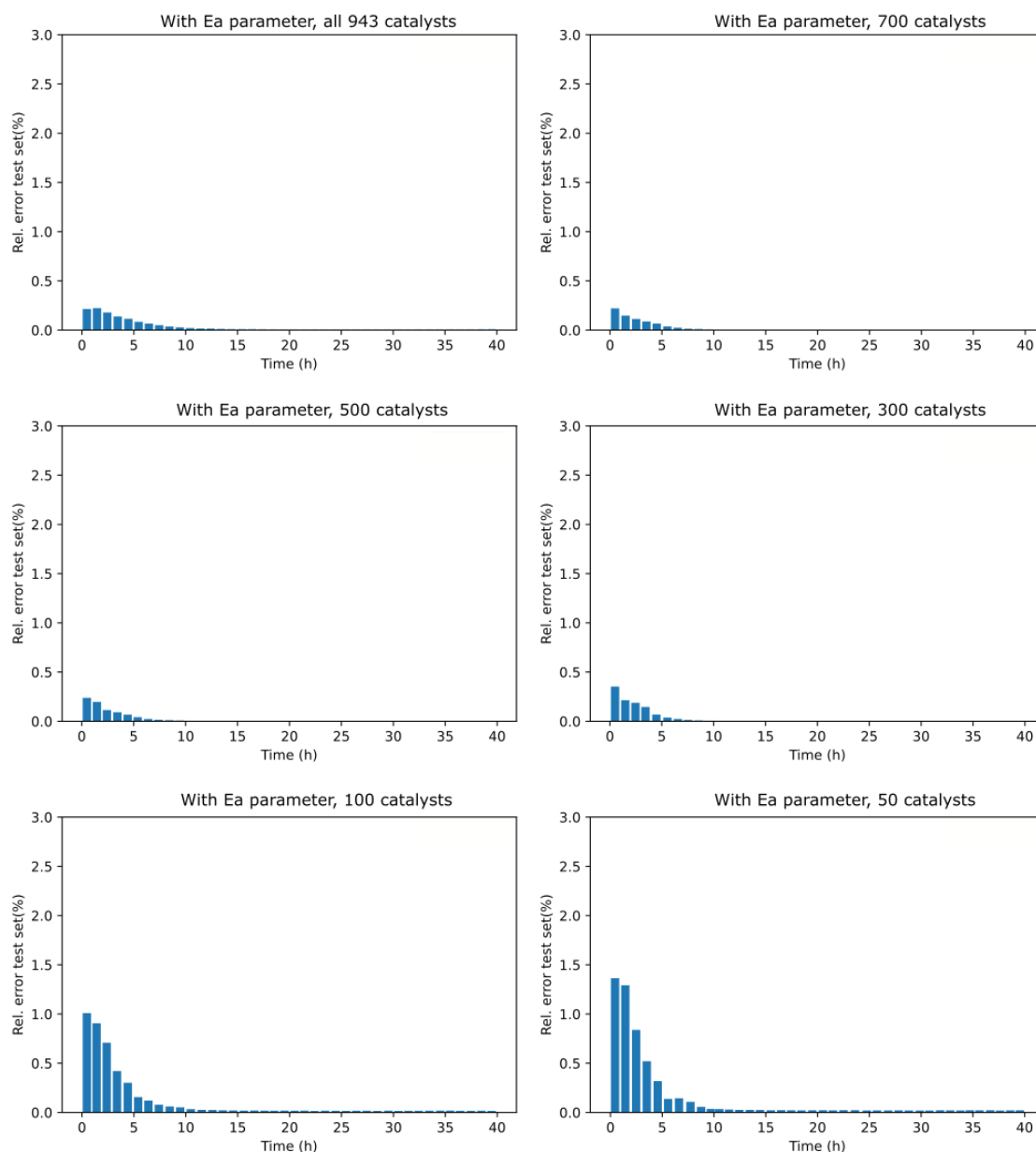


set of features that correlate with the Ea parameter are annotated with "With features" whereas the models that were directly provided with the Ea parameter are annotated with "With Ea parameter". As expected the resulting relative error is much lower than the dataset with the set of features that correlate with the Ea parameter. The improved performance can be attributed to the removed layer of complexity that is the structure-activity relation between the set of features and the Ea parameter.

Unlike the results chart for the datasets that include the set of features (figure 4.7), figure 4.9 displays the expected trend in the influence of the database size. The difference between these results can be explained by the time dependence of the relative error. As is shown in figure 4.10, the errors for predicting the equilibrium concentrations are equally small across the varying database sizes, which results in the absence of the small-error stacking effect and thereby a clear pattern in the influence of the database size in figure 4.9.



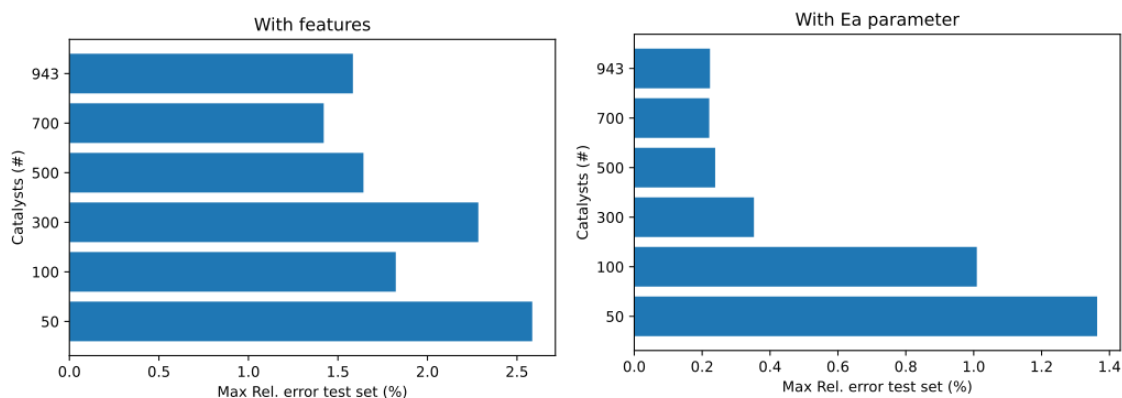
**Figure 4.9:** The ML models' test set performances with the inclusion of the Ea parameter, across the database sizes



**Figure 4.10:** A view of the time dependence of the test set performances with the inclusion of the Ea parameter, across the database sizes

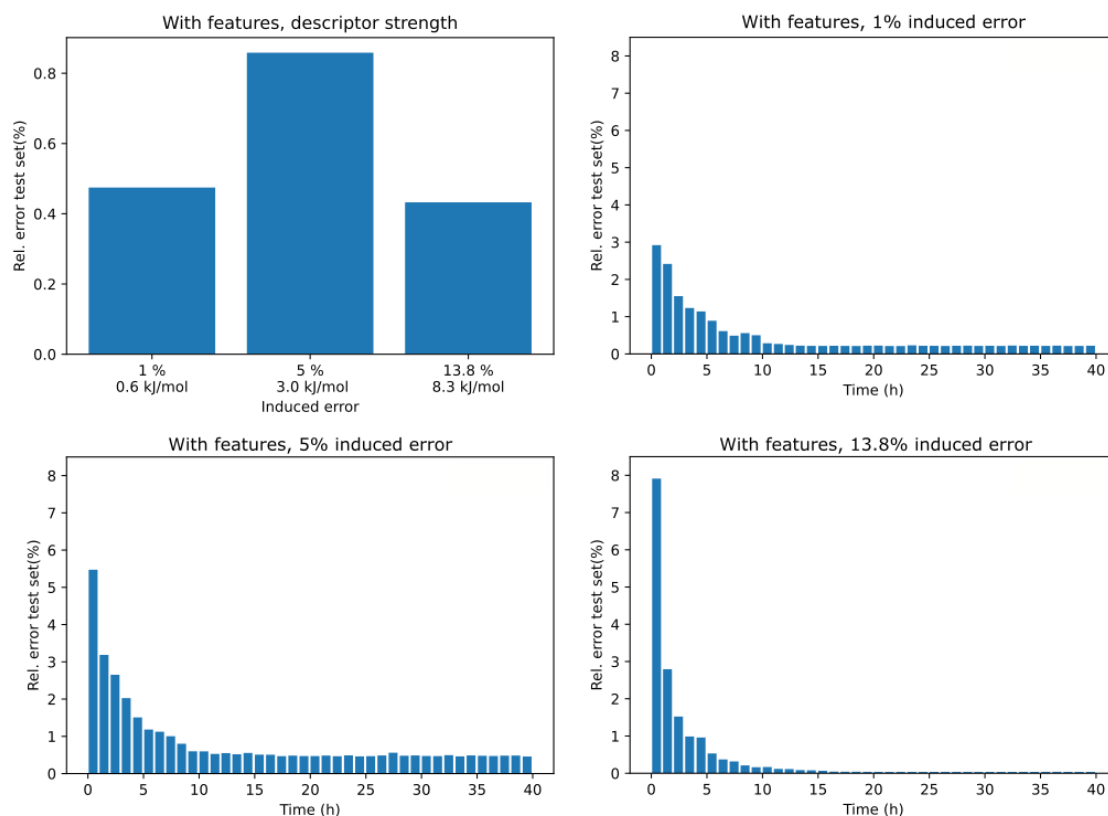
The maximum relative error between the 40 time intervals for each of the database sizes is summarized in figure 4.11. The models that were provided with the Ea parameter show a significantly lower error than those that were provided the set of features. Especially the Ea parameter models that were trained on datasets with 500 and more catalysts are considerably more accurate. It is noteworthy that the error across these three ML models is almost constant. This means that the database that comprises 500 catalyst already contains enough data points for the ML algorithm to understand the underlying mechanisms, which is the relation between the Ea parameter and the transient composition data, i.e.

the kinetic model. When decreasing the number of catalysts in the database it is shown that around 300 catalysts the database size starts limiting the degree to which the underlying principles can be learned by the ML algorithm. For the ML models that were trained on the set of features this pattern is less recognizable. This result as well as the increase in the error can only be caused by the added layer of complexity between the model's features and the transient composition that, i.e. the structure-activity relation.



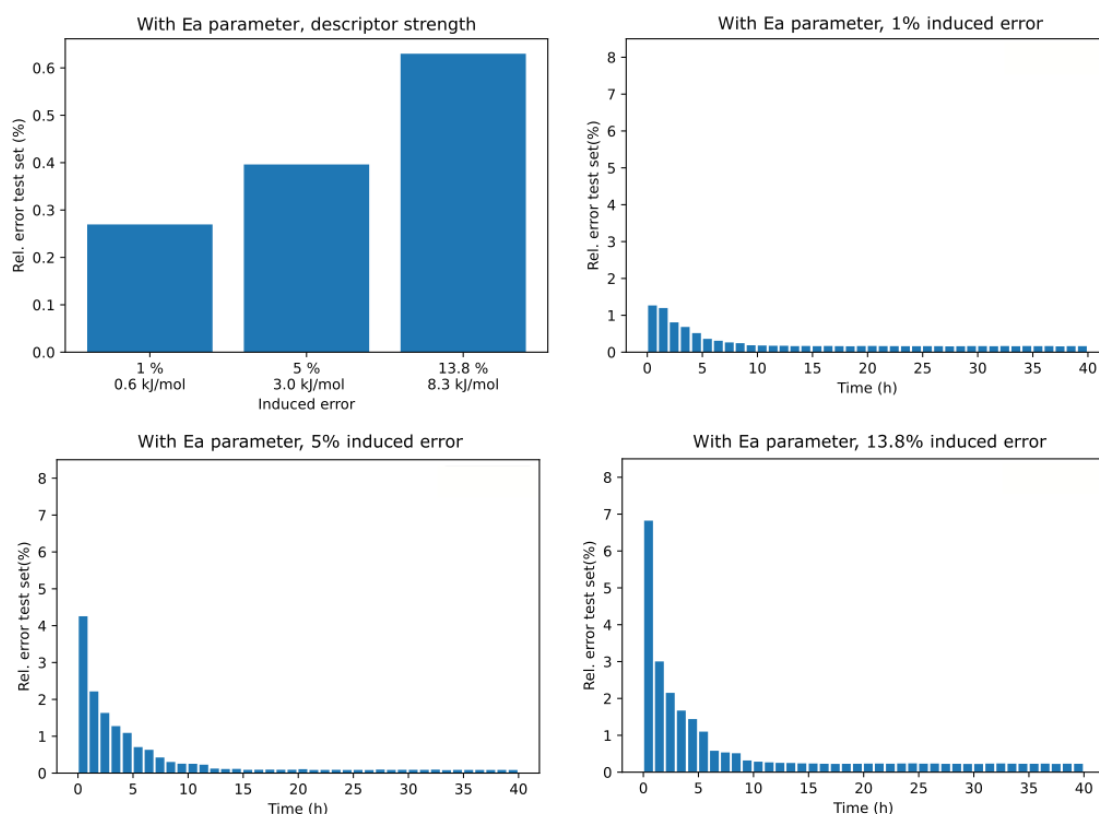
**Figure 4.11:** A comparison between the database size dependent performances in terms of maximum relative error of the ML models that exclude and include the Ea parameter in their respective training data

The results for the descriptor strength are similar to those of the database size variations in the sense that the expected pattern is not clearly visible in the relative error. It is self-evident that inducing an uncertainty to the relation between the features and the Ea parameter should result in a decrease in performance. However, the top left chart of figure 4.12 fails to show this trend. Again the time dependent results reveal that the small-error stacking effect is the reason for this misrepresentation. The time dependent results in figure 4.12 clearly show that the larger relative errors indeed arise from higher induced errors, which correspond to a lower descriptor strength.



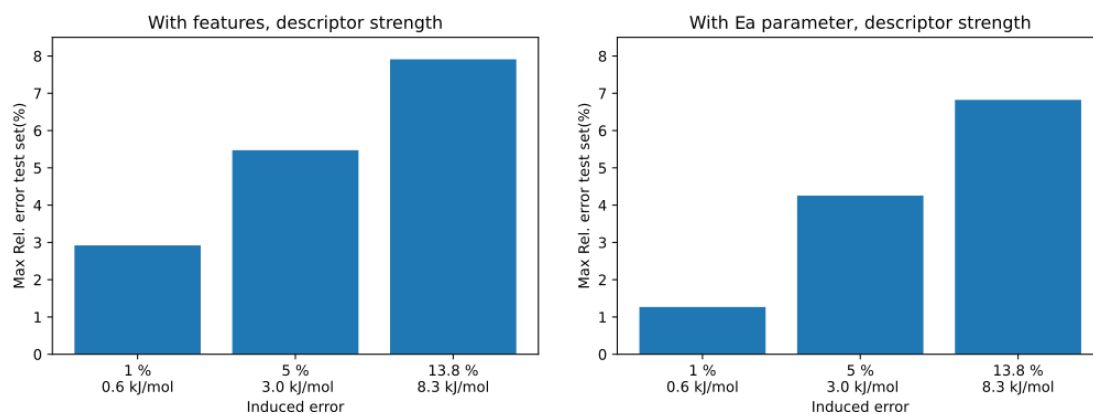
**Figure 4.12:** The ML models' test set performance with varying descriptor strengths

In analogy with the database size results, the small-error stacking effect is less pronounced for the descriptor strength when the  $E_a$  parameter is included as a feature. Since a similar pattern was observed for the database size variations it is likely that the disparity between the predictive capability for the equilibrium concentrations is caused by the added complexity that comes from the relation between the features and the  $E_a$  parameter relation.



**Figure 4.13:** The ML models' test set performance with the inclusion of the Ea parameter, across the descriptor strengths

The maximum relative errors from the time dependent results are summarized in figure 4.14. This figure is able to clearly present the expected trend that should arise from weakening the descriptor strength. In addition, these errors are relatively large compared to the errors arising from varying the database size. The large influence of the descriptor strength was to be expected because the uncertainties in the Ea parameter propagate exponentially to the transient composition data.



**Figure 4.14:** A comparison between the descriptor strength dependent performances in terms of maximum relative error of the ML models that exclude and include the Ea parameter in their respective training data

In summary, the standard evaluation method did not immediately provide insight in the true effects of the dataset variations. The main cause was the inflated influence of insignificant errors that we defined as small-error stacking. However, by assessment of the relative error over time we were able to reveal the true trends in the results. Next to displaying the errors' patterns, the standard method of evaluation also quantified the average relative errors that can be expected in prediction of transient composition data. The main shortcoming of the standard evaluation method relate to inquiries about the application of the ML models in catalyst discovery. An example is the inability of the standard evaluation method to determine the appropriate database size that would be necessary in applying the ML models in catalyst discovery. Additionally, this method provides no insight in the effects of the dataset variations on the integrity of measures for catalytic activity that are inferred from the predicted concentrations. The secondary evaluation method aims to complement the standard method by providing these necessary insights.

In this applicative evaluation method, the concentration profile for 20 new catalysts is predicted. The charts shown in figure 4.15 and 4.16 show the influence of the database size on the assessment of catalytic behaviour through the IRR and FC respectively. For each of the 20 evaluation catalysts the true value for these parameters is shown along with the value that is calculated from the predicted concentration points. Both figures show that the database sizes of 943, 700, and 500 catalysts are able to correctly predict the most promising catalysts. In addition, these ML models are also able to correctly predict which catalysts would be unrewarding to develop. It should be noted that the predicted values are not flawless and larger errors do occasionally lead to unjustly favoring of the wrong catalyst. For example the ML model that trained on a database with 500 catalysts shows an overestimated catalyst such that it would be favored over some better performing catalysts. These problematically large errors in predicting catalytic activity are more frequent and severe with smaller datasets. The predicted catalytic activity parameters from the ML model that is trained on a dataset of

300 catalysts show large deviations from the true catalytic activity parameters and even falsely predicts that one of the worst catalysts would be among the most promising catalysts. The remaining ML models which are trained on datasets consisting of 100 and 50 catalysts produce errors in predicting the transient composition data to the extent that the inferred catalytic activity parameters are unreliable. The desired performance of the ML models is only consistently achieved by the models that were trained on data of at least 500 catalysts. Therefore, these results indicate that future databases that aim to use ML for catalyst discovery in a similar manner as in this case study, should contain data of at least 500 catalysts. This is a very large number, and considerable efforts are required in constructing a kinetic database of this size. Despite the result that ML models trained on smaller databases are not very reliable, these ML models are able to predict catalytic performance to some extent, and could still be useful in catalyst discovery considering the speed at which these predictions can be made.

Moreover, the essence of these results show that statistically inferring the most promising catalysts is close to impossible when only having access to datasets of less than 100 catalysts. Within the current state of catalysis researchers rarely examine the performance of over 100 catalysts for a particular reaction, therefore one could argue that the information gathered through the current catalytic research method is likely insufficient to fully understand the underlying mechanisms in catalysis, and that the most promising catalysts are still to be discovered.

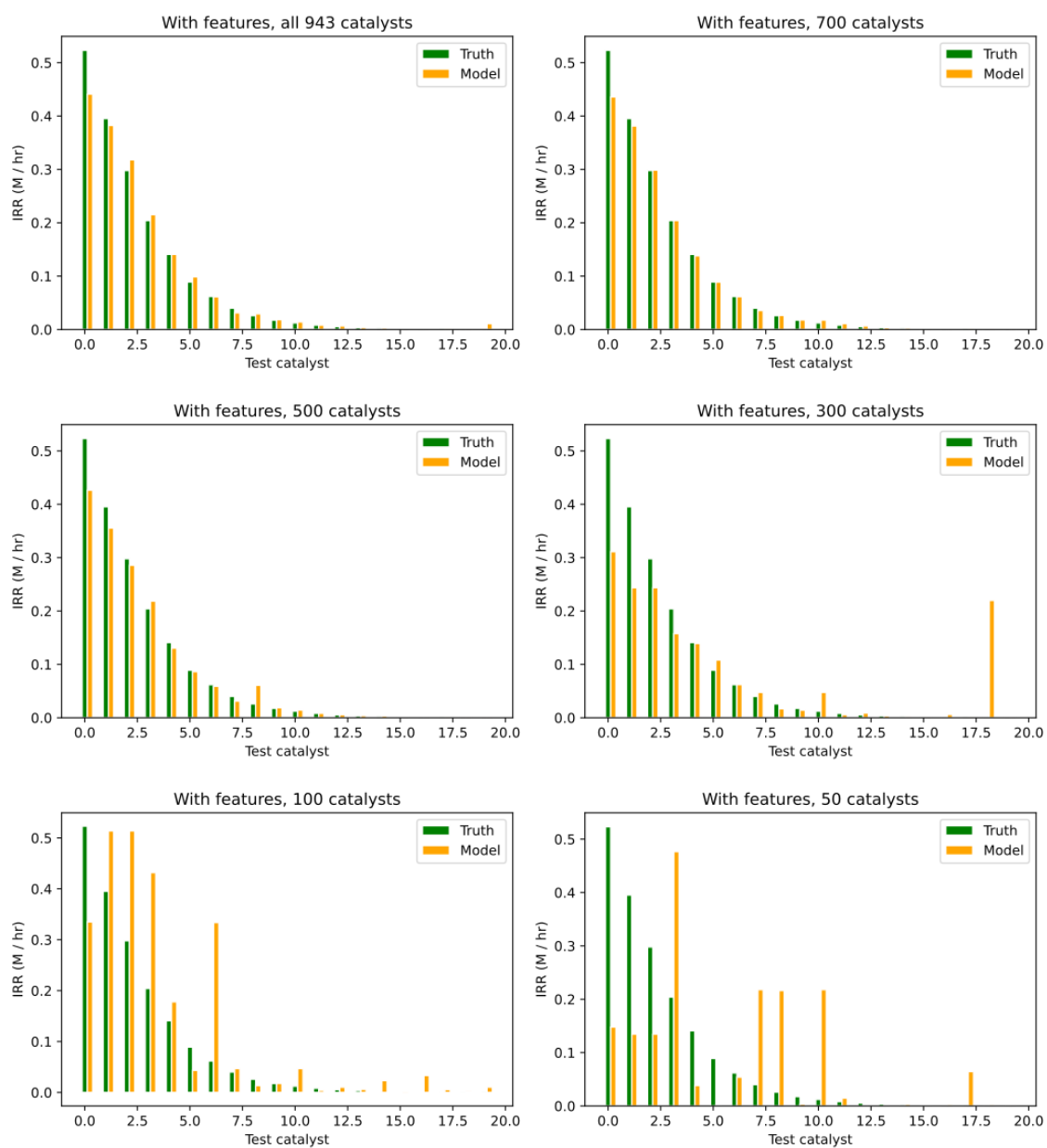
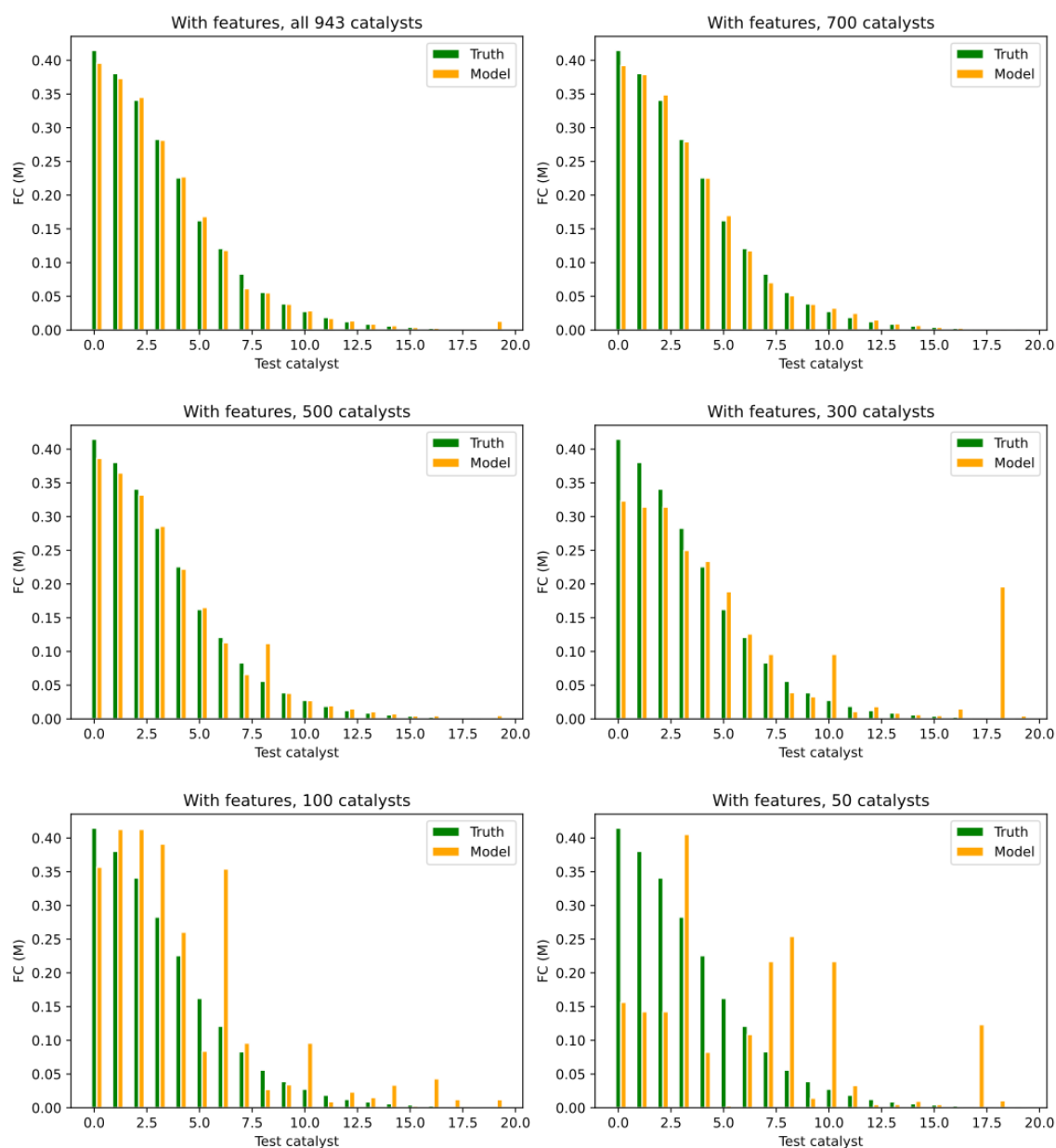


Figure 4.15: The influence of the database size on IRR prediction

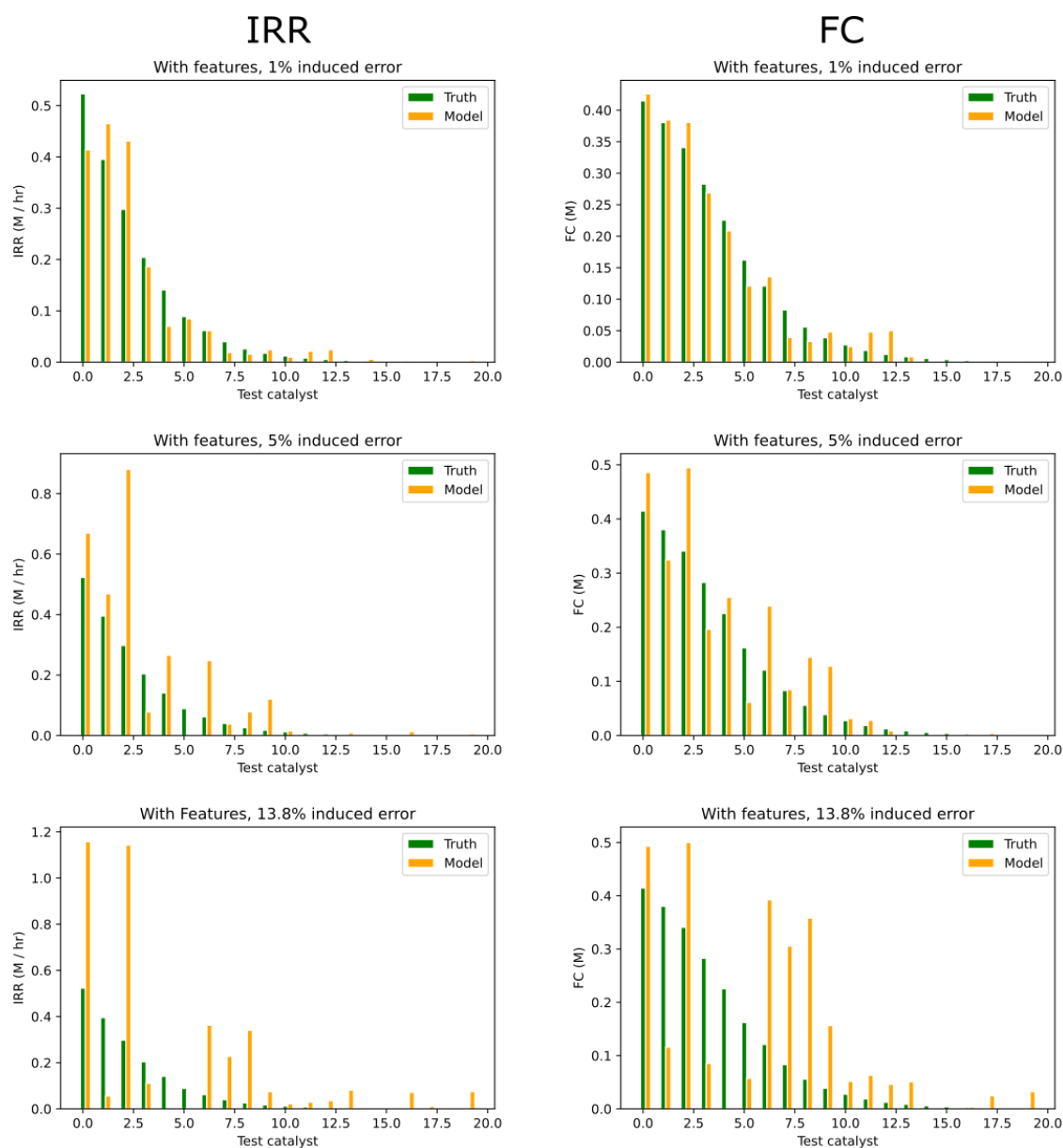




**Figure 4.16:** The influence of the database size on FC prediction

The applicative method of evaluation was also used to study the effect of the descriptor strength in predicting the catalytic performance. In agreement with the results from the standard evaluation method, figure 4.17 shows that the descriptor strength has a large impact on the ML models' ability to predict the catalytic performance. At an induced error of 1% to the relation between the features and the  $E_a$  parameter, the deviations from the true catalytic activity metrics are already pronounced. Despite these large differences, the ML model is able to predict the most promising catalyts while also correctly predicting the catalytic performance of the unsuitable catalyts. The results for an induced uncertainty of 5% show that there is some capability in predicting catalytic performance but the large errors make

these predictions unreliable. This unreliability is increasingly pronounced in the results for the ML models that are trained on data with a 13.8% induced uncertainty. These results show that having a large descriptor strength in the dataset is vital for achieving ML applications in catalyst discovery. The influence of the descriptor strength shows that the main concern in future database construction in catalysis will likely be the data quality instead of the database size.



**Figure 4.17:** The influence of the descriptor strength on prediction of the catalytic performance

To conclude, this case study investigated the potential of ML in catalyst discovery through an artificial dataset. It should be noted that a single case study is incapable of representing every ML endeavor in catalysis informatics and the applicability of this case study to future works should be judged for each case individually. In this case study, ML models were trained to predict transient composition data

from molecular descriptors. Through dataset variations, we quantified the effect of data quality and quantity on both the models' predictive performance and their potential of being applied in ML aided catalyst discovery. Finally, the results of this case study provide direction in determining the objectives for future research in database construction from real catalytic data.

# 5

## CONCLUSION

The conclusions from this thesis project can be divided into two parts. The first part involves the development of a data management and analytics platform, and the second part involves the application of this platform through a case study. During the development of the platform, our main priority was to provide accessibility for catalysis researchers to ML. This is realized by guiding the researcher through the ML workflow without requiring prior coding experience or ML proficiency. In addition, our platform serves as a gateway to interact with a suitable database instance, that can account for the diversity in catalytic research data. As a result, we present a platform where the researcher can manage their data as well as gain insight from their data through the construction of effective ML models.

In the second part of this thesis, the platform's potential is demonstrated by means of a case study, where ML models are built to predict the catalytic performance based on molecular descriptors of the catalyst. In this case study an artificial database was constructed because of the lack of available kinetic databases. The final artificial database contains 943 catalysts, where molecular descriptors are coupled to time-dependent species concentrations. By making variations of the artificial dataset we studied the influence of database size and descriptor strength on the performance of ML models. The results showed that including high quality descriptors is extremely valuable in the prediction of catalytic performance, and we believe that research concerned with finding suitable descriptors will be instrumental in future catalytic database construction. In addition, the results quantified the database sizes that led to ML models with reliable and unreliable predictive capability for our case study. The gathered insights of this thesis can prove useful in determining the setup of future experiments, as well as determining the requirements for future catalytic datasets. Thereby, this case study provides directions for future endeavours towards automated catalyst discovery.

## 5.1. OUTLOOK

The developed platform is still a prototype, and has yet to incorporate feedback from user experiences. Only after repeated cycles of user evaluation and platform refinement, we can truly judge its effectiveness in making ML more accessible in catalysis research. An example of a refinement step towards a finished platform is the design of the pages. During the development of the platform the design of the pages was focused around their practicality, and can be improved in terms of clarity and aesthetics.

Furthermore, creating valuable databases still requires a considerable effort from researchers to create uniformity in their data. In this regard, the development of an ontology will be beneficial, and can contribute to the construction of larger catalytic databases.

Finally, the platform can benefit from including optimization algorithms for the constructed ML models. For catalytic performance models similar to those constructed in the case study, the optimization can provide direct recommendations for future experiments, which could even be used in automation of an experimental workflow.

# 6

## ACKNOWLEDGEMENTS

I would like to express my gratitude towards my daily supervisor Robbert van Putten for the great guidance, advice, and support, which allowed my thesis project to be very instructional and exciting.

Also, I am grateful to have been part of the innovative ISE research group, where I have had the opportunity to participate in the insightful discussions with my head supervisor Prof. Dr. Evgeny Pidko and other members of the group.

Finally, I would also like to thank my parents whose continuous support made the difficult challenges of my project easier to overcome.

## BIBLIOGRAPHY

- [1] Osuna, E.; Freund, R.; Girosit, F. Training support vector machines: an application to face detection. Proceedings of IEEE computer society conference on computer vision and pattern recognition. 1997; pp 130–136.
- [2] Patel, J.; Shah, S.; Thakkar, P.; Kotecha, K. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert systems with applications* **2015**, *42*, 259–268.
- [3] Chen, C.; Seff, A.; Kornhauser, A.; Xiao, J. Deepdriving: Learning affordance for direct perception in autonomous driving. Proceedings of the IEEE international conference on computer vision. 2015; pp 2722–2730.
- [4] Zwanzig, R.; Szabo, A.; Bagchi, B. Levinthal’s paradox. *Proceedings of the National Academy of Sciences* **1992**, *89*, 20–22.
- [5] Jumper, J.; Evans, R.; Pritzel, A.; Green, T. High Accuracy Protein Structure Prediction Using Deep Learning. <https://deepmind.com/blog/article/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology>.
- [6] Bligaard, T.; Bullock, R. M.; Campbell, C. T.; Chen, J. G.; Gates, B. C.; Gorte, R. J.; Jones, C. W.; Jones, W. D.; Kitchin, J. R.; Scott, S. L. Toward benchmarking in catalysis science: best practices, challenges, and opportunities. *Acs Catalysis* **2016**, *6*, 2590–2602.
- [7] Medford, A. J.; Kunz, M. R.; Ewing, S. M.; Borders, T.; Fushimi, R. Extracting knowledge from data through catalysis informatics. *ACS Catalysis* **2018**, *8*, 7403–7429.
- [8] Ashburner, M.; Ball, C. A.; Blake, J. A.; Botstein, D.; Butler, H.; Cherry, J. M.; Davis, A. P.; Dolinski, K.; Dwight, S. S.; Eppig, J. T., *et al.* Gene ontology: tool for the unification of biology. *Nature genetics* **2000**, *25*, 25–29.
- [9] White, J. M.; Bercaw, J. Opportunities for Catalysis in The 21st Century. A report from the Basic Energy Sciences Advisory Committee.
- [10] Gorse, A.-D. Diversity in medicinal chemistry space. *Current topics in medicinal chemistry* **2006**, *6*, 3–18.

- [11] Coley, C. W.; Green, W. H.; Jensen, K. F. Machine learning in computer-aided synthesis planning. *Accounts of chemical research* **2018**, *51*, 1281–1289.
- [12] Cherkasov, A.; Muratov, E. N.; Fourches, D.; Varnek, A.; Baskin, I. I.; Cronin, M.; Dearden, J.; Gramatica, P.; Martin, Y. C.; Todeschini, R., *et al.* QSAR modeling: where have you been? Where are you going to? *Journal of medicinal chemistry* **2014**, *57*, 4977–5010.
- [13] Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: a benchmark for molecular machine learning. *Chemical science* **2018**, *9*, 513–530.
- [14] Van Santen, R. A.; Neurock, M.; Shetty, S. G. Reactivity theory of transition-metal surfaces: a Brønsted- Evans- Polanyi linear activation energy- free-energy analysis. *Chemical reviews* **2009**, *110*, 2005–2048.
- [15] Nørskov, J. K.; Abild-Pedersen, F.; Studt, F.; Bligaard, T. Density functional theory in surface chemistry and catalysis. *Proceedings of the National Academy of Sciences* **2011**, *108*, 937–943.
- [16] Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Big data meets quantum chemistry approximations: the  $\Delta$ -machine learning approach. *Journal of chemical theory and computation* **2015**, *11*, 2087–2096.
- [17] Nandy, A.; Duan, C.; Janet, J. P.; Gugler, S.; Kulik, H. J. Strategies and software for machine learning accelerated discovery in transition metal chemistry. *Industrial & Engineering Chemistry Research* **2018**, *57*, 13973–13986.
- [18] Fujima, J.; Tanaka, Y.; Miyazato, I.; Takahashi, L.; Takahashi, K. Catalyst Acquisition by Data Science (CADS): a web-based catalyst informatics platform for discovering catalysts. *Reaction Chemistry & Engineering* **2020**, *5*, 903–911.
- [19] Toyao, T.; Maeno, Z.; Takakusagi, S.; Kamachi, T.; Takigawa, I.; Shimizu, K.-i. Machine learning for catalysis informatics: recent applications and prospects. *ACS Catalysis* **2019**, *10*, 2260–2297.
- [20] Raccuglia, P.; Elbert, K. C.; Adler, P. D.; Falk, C.; Wenny, M. B.; Mollo, A.; Zeller, M.; Friedler, S. A.; Schrier, J.; Norquist, A. J. Machine-learning-assisted materials discovery using failed experiments. *Nature* **2016**, *533*, 73–76.
- [21] Hummelshøj, J. S.; Abild-Pedersen, F.; Studt, F.; Bligaard, T.; Nørskov, J. K. CatApp: a web application for surface chemistry and heterogeneous catalysis. *Angewandte Chemie International Edition* **2012**, *51*, 272–274.
- [22] Winther, K. T.; Hoffmann, M. J.; Boes, J. R.; Mamun, O.; Bajdich, M.; Bligaard, T. Catalysis-Hub.org, an open electronic structure database for surface reactions. *Scientific data* **2019**, *6*, 1–10.
- [23] El Naqa, I.; Murphy, M. J. *machine learning in radiation oncology*; Springer, 2015; pp 3–11.



- [24] Awad, M.; Khanna, R. *Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers*; Apress: Berkeley, CA, 2015; pp 1–18.
- [25] Electron Runtime framework. <https://www.electronjs.org/>, Accessed: 2020-12-31.
- [26] Jiscoot, N. Data management systems for catalysis. Bachelor's Thesis, Delft University of Technology, Delft, The Netherlands, 2020.
- [27] MongoDB, Who uses MongoDB? <https://www.mongodb.com/who-uses-mongodb/>, Accessed: 2020-12-31.
- [28] Tableau Academic Programs. <https://www.tableau.com/academic/>, Accessed: 2020-12-31.
- [29] Scikit-Learn Testimonials. <https://scikit-learn.org/stable/testimonials/testimonials.html/>, Accessed: 2020-12-31.
- [30] Zöllner, M.-A.; Huber, M. F. Benchmark and survey of automated machine learning frameworks. *arXiv preprint arXiv:1904.12054* **2019**,
- [31] Orlenko, A.; Moore, J. H.; Orzechowski, P.; Olson, R. S.; Cairns, J.; Caraballo, P. J.; Weinsilboum, R. M.; Wang, L.; Breitenstein, M. K. Considerations for automated machine learning in clinical metabolic profiling: Altered homocysteine plasma concentration associated with metformin exposure. *PSB*. 2018; pp 460–471.
- [32] Jupyter. <https://jupyter.org/>, Accessed: 2020-12-31.
- [33] Günay, M. E.; Yildirim, R. Knowledge extraction from catalysis of the past: a case of selective CO oxidation over noble metal catalysts between 2000 and 2012. *ChemCatChem* **2013**, *5*, 1395–1406.
- [34] Zavyalova, U.; Holena, M.; Schlögl, R.; Baerns, M. Statistical analysis of past catalytic data on oxidative methane coupling for new insights into the composition of high-performance catalysts. *ChemCatChem* **2011**, *3*, 1935–1947.
- [35] Odabaşı, Ç.; Günay, M. E.; Yildirim, R. Knowledge extraction for water gas shift reaction over noble metal catalysts from publications in the literature between 2002 and 2012. *International journal of hydrogen energy* **2014**, *39*, 5733–5746.
- [36] Bronsted, J. Acid and Basic Catalysis. *Chemical Reviews* **1928**, *5*, 231–338.
- [37] Wang, S.; Petzold, V.; Tripkovic, V.; Kleis, J.; Howalt, J. G.; Skulason, E.; Fernández, E.; Hvolbæk, B.; Jones, G.; Toftelund, A., *et al.* Universal transition state scaling relations for (de) hydrogenation over transition metals. *Physical Chemistry Chemical Physics* **2011**, *13*, 20760–20765.
- [38] Singh, A. R.; Rohr, B. A.; Gauthier, J. A.; Nørskov, J. K. Predicting chemical reaction barriers with a machine learning model. *Catalysis Letters* **2019**, *149*, 2347–2354.