Streamlining Data Integration and Decision Support in Refinery Operations

Ocan Şahin[a], Aslı Yasmal[b], Mustafa Oktay Samur[c] and Gizem Kuşoğlu Kaya[d]
Turkish Petroleum Refinery, Körfez, Kocaeli, 41780, Turkey

Keywords: Digital Twins, Process Optimization, Modelling, Simulation and Architecture, Real-Time Analysis, Decision Support Systems.

Abstract: Refineries, operating with millions of dollars at stake, face significant economic consequences even with just 30 minutes of non-ideal operation. To address this challenge, this paper presents an industrial application of seamless integration of two different data sources into a complicated decision support tool that enables feedforward decisions. The integration is done in Node-RED, facilitating the data flow from two sources leveraging SOAP calls and COM interfaces in Python to automate the model manipulation, thus generating live estimates before operation takes place. A dashboard is developed, provides a user-friendly interface for visualizing the data and making informed decisions on how to increase efficiency and feed the existing model predictive control architecture. This use-case demonstrates the effectiveness of Node-RED in streamlining data integration, automation, and decision-making processes in industrial settings is demonstrated, contributing to improved operational efficiency and profitability in the refinery industry.

1 INTRODUCTION

The refining industry, being one of the oldest, stands to gain greatly from advancements in technology through automation and data science in the last decades. Cracking and treatment units in an oil and gas refinery are crucial to refining process, where minor optimizations in their operation can lead to improvements in overall efficiency and output. A practical approach to improving the process would be to focus on the final outcome and adjust the operational parameters accordingly (Yasmal et al., 2022; Kaya et al., 2023). One key process we focus on is the Diesel Hydro Processing (DHP) unit, which is critical due to its role in the catalytic conversion of a naphtha and diesel mixture. The DHP unit plays an essential part in the reactors, where this conversion occurs, making it a central element in optimizing the overall process. In this context, diesel hydroprocessing is an important refining process that consists of hydrodesulphurization to remove the unwanted sulfur from the diesel cut. The process consists of hydrocracking and hydrotreating to produce a diesel product with the required characteristics (Aydın et al., 2015). Related to this, the plant model comprises two distinct parts: hydrodesulphurization (HDS) (Kabe et al., 1999) and hydrocracking (HC) (Ward, 1993). In accordance with BS EN regulation (Automotive fuels, 2023), the use of the HDS process is a key factor in achieving a product with superior cleanliness and ultra-low sulfur content, eliminating all negative environmental impacts such as sulfur dioxide emissions and water pollution (Safari & Vesali-Naseh, 2018). The hydrocracking process breaks down hydrocarbon molecules into lower molecular weight carbon chains. This is a high temperature process, around 650K to 700K (Park et al., 2018), with strong dependency on its catalysis' performance and very difficult to optimize due to being a black box. When done correctly, it can convert fuels into high-value products with a high hydrogen/carbon ratio and low metal contaminants within one catalysis life cycle (Rana et al. 2007).

[a] https://orcid.org/0000-0002-5911-8076
[b] https://orcid.org/0000-0002-8080-5591
[c] https://orcid.org/0009-0005-0931-4190
[d] https://orcid.org/0000-0003-2825-0143

Şahin, O., Yasmal, A., Samur, M. and Kaya, G.
Streamlining Data Integration and Decision Support in Refinery Operations.
DOI: 10.5220/0012950500003822

In *Proceedings of the 21st International Conference on Informatics in Control, Automation and Robotics (ICINCO 2024) - Volume 2*, pages 403-409
ISBN: 978-989-758-717-7; ISSN: 2184-2809

The aim of this study is to establish links between a substantial quantity of data sources, implement live estimation models in an automated manner, and devise an interactive decision-making assistive mechanism to the control of hydrocracking reactor and separation units. The written manuscript is structured as outlined below: Section 1, problem description Section 2 presents the methods, system design, representational state transfer, application programming interface, historian database server, and hardware implementation. In Section 3, the performance and advantages of the experiment, and the experimental results are described in detail. Section 4 includes a brief discussion about operational efficiency, technical and economic benefits, and potential future directions. Finally, Section 5 concludes the whole study and provides potential future research directions.

2 PROBLEM DESCRIPTION

The DHP, which is the plant this study focuses on, while producing LPG, naphtha as side products, mainly produces vast amounts of clean and cracked diesel. It should be noted that highly exothermic reactions occur in the reactors. Due to the substantial presence of hydrogen in the system, the highly exothermic reaction is mitigated through the introduction of cooling hydrogen quenches. This cooling process necessitates additional hydrogen to be fed to the circulation because hydrogen is consumed. Simultaneously, catalytic reactivity is maximized to maintain reactor temperatures above a certain threshold, as higher temperatures enhance reactivity. However, this cannot be pushed to the extreme, because higher temperatures not only increase reactivity, but it also favors multiple processes that reduce catalysis life cycle such as faster coke deactivation or support sintering (Gruia, 2006). It is a delicate balance that involves optimizing the cracking process, ensuring reactor safety, extending catalyst lifespan, and managing the optimum hydrogen ratio. This equilibrium, while crucial for the operation's success and profitability, is complex to maintain, but the challenges that it brings with it are also significant

For instance, the plant can encounter issues with off-spec diesel product accumulating at the base of the stripper column, resulting in significant product mismatches. In addition to that, it can also encounter significantly over-cracked diesel, which means the catalyst and the reactors are unnecessarily overworked. To mitigate these issues and maintain the high quality of diesel produced, it is essential to implement a feed-adaptive control system. This is especially important because the characteristics of the incoming diesel feed often vary, necessitating various operating conditions to consistently achieve the desired results.

In response to these problems, the application described in this work presents a decision support tool for estimating the remaining operational life of major plant components. By offering this prediction, the tool allows plant personnel to solve potential issues before they have an impact on the final product's profitability. This proactive strategy ensures that the facility performs optimally and produces high-quality products.

3 METHODOLOGY

This study's methodology focuses on integrating real-time operational data with predictive modeling and simulation tools to improve decision-making capabilities in refinery operations. To accomplish this, we have implemented a multi-faceted approach involving live data extraction, predictive modeling using hydroprocessing and separation simulations, and a flow-based architecture for data handling and model integration. The system leverages a combination of SOAP API for secure and structured data retrieval, MATLAB for hydroprocessing estimations, commercial simulation software for separation modeling, and Node-RED for orchestrating data flows and presenting live results on a user-accessible dashboard. This integrated framework supports proactive adjustments in processing operations, aligning closely with the dynamic requirements of refinery environments to improve operational efficiency and reduce potential risks. The complete flow diagram of the solution can be seen in Fig. 1.

3.1 Data Connections

Continuous operation required a continuous solution that can support the decisions made live, so the challenge ahead was obvious. The first step was to feed live operational data into accurate models that run faster than the decision support requirements. These models are hydroprocessing estimation model enhanced in MATLAB and separation simulation models created using commercial simulating software. To achieve that goal, we opted to use an application programming interface (API) that utilizes Simple Object Access Protocol (SOAP) calls. The

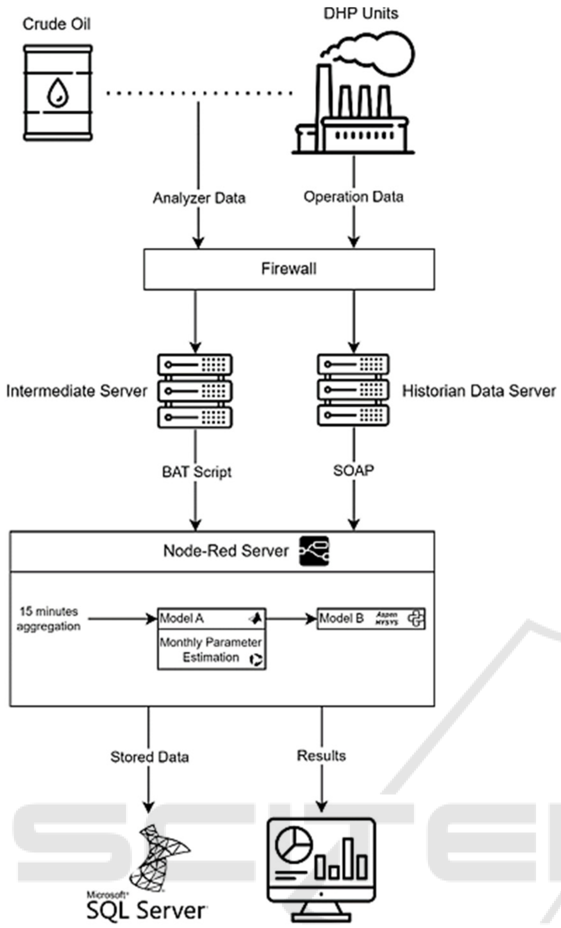

Figure 1: Flow Diagram of the complete application.

SOAP API is a widely used protocol for exchanging structured information in web services. In the context of our application, the SOAP API serves as a communication bridge between the application developed and the historian database server (World Wide Web Consortium, 2010). As a large enterprise we are using SOAP instead of RESTful APIs because SOAP provides a more structured and standardized approach that aligns well with our complex enterprise requirements (Fielding, 2000). SOAP's reliance on XML ensures consistent data representation, which is crucial for our integration with diverse systems. Additionally, SOAP's support for various transport protocols allows us to seamlessly communicate with different platforms within our enterprise architecture.

To retrieve the required data, the SOAP request is constructed from an xml-based envelope and the secure server parses the envelope, identifies the action and its parameters, runs the query and prepares response. The response contains the requested data prepared by using the parameters supplied, such as operation data point tag name of the operation data point in Table 1. (e.g.: 10TIC5.PV), tag name prefix represents the plant number, this table consists tags from plant 10 and plant 15. A temperature controller process value (PV) on first row. A flow controller's set point (SP) on second row. A pressure indicator's process value (PV) on third row and the same pressure indicator's respective valve opening (OP) on fourth row. Confidence represents the sureness of the collected value and very rarely reads something other than 100 or 0 (100 represents correct, 0 represents incorrect values). Due to data privacy, we cannot share real operational data.

Table 1: Dummy Sample Plant 10 and 15 data response.

Timestamp	Tag Name	Value	Confidence
10.01.2023T12:30:00	10TIC5.PV	35°C	100
10.01.2023T12:30:00	10FIC2.SP	500 m^3/h	100
10.01.2023T12:30:00	15PI12.PV	12 bar	100
10.01.2023T12:30:00	15PI12.OP	25%	0

To achieve the level of predictive control we needed, feeding live operational data into our models was not enough. A traditional feedback loop makes adjustments in response to system output, which can cause delays and inefficiencies. The goal was a feed-forward system that would proactively adjust on the basis of input data before problems occurred. Hence, an online analyzer was installed to the input feed of the system to model the chemical properties of the incoming liquid. Analyzer uses Near-Infrared (NIR) (Falla et al., 2006) spectroscopy technology. It estimates feed total boiling point using the internally developed statistical models and sends the data to its own on-site computer. From behind the firewall, a trivial bat script writes the data to an intermediate server that has one-directional communication with the analyzer computer and to our solution server. Although the analyzer can take measurements once every couple of seconds, it is set to work once every five minutes. Results are sent to the solution server within a similar frequency. This is due to the time required to run the other models reliably. There was no need to create more input data if the models cannot run fast enough.

3.2 Hydro Processing Estimation Model

The solution models' first part consists of distinct hydrodesulphurization calculations and hydrocracking calculations. While we are modelling the reactor, due to the different bed types and bed lengths of the reactors, each bed has been considered as a separate reactor in series and calculations have been made accordingly. In the HDS reactor, sulfur compounds are removed from the feed based on the sulfur specification. Cracking of the feed carries out in the HC reactor. One of the underlying purposes of this study is to keep the boiling point of the product within the desired values of the unit. In the MATLAB part (The MathWorks Inc., 2022), models are constructed to predict the composition of reactor exit and reactor bed exit temperatures. Certain properties of the feed are used as inputs. These include TBP values at different temperatures, flow rate, sulfur content, and bed inlet temperatures. During the calculations, some assumptions are made for both HDS and HC reactions. In the HDS beds, it is assumed that the formation and effects of Hydrogen Sulfide (H_2S) are ignored, and no cracking reactions occur. For both reactors, it is assumed that reactions are adiabatic, homogeneous, and liquid phase, reactions are first order, heat capacities of components are constant, and activation energies for the beds are constant. For this estimation, feed properties are used, and the pseudo-true boiling point is calculated.

Feed characterization data is taken from the summary day of each month. Summary day is a special day where extra examples are taken from the plant to be analysed in the laboratory, like an offline snapshot of the operation. Using this data, the cost function is tried to be minimized. The cost function for optimizing the kinetic parameters considers the reactor bed outlet temperatures and the weight fractions of the reactor effluent. Thus, optimum parameters are determined through an iterative study for each month.

The models are fed 32 different tags, including the estimated incoming feed characteristics combined with the current operating conditions of the plant. From these 32 tags, 10 represents the incoming feed characteristics; however, the other 22 tags are selected after careful field tests to encapsulate the maximum amount of operational meaning while using the least amount of tag load possible. They are pulled for the last 15 minutes of operation and the mean values of the 15 minutes are used in our models. The model successfully estimates the creation of the major products of the plant, but because of the black box nature of the system, results cannot be validated until the products are separated from each other. From the first model results, only the reactor temperatures are something that can be meaningful to use, the cracked diesel compositions consist of 145 features cannot be used anywhere before second model runs. These temperature values can be further used to fine-tune the overall solution in an iterative way to find the optimal operating conditions for the desired outcome.

The MATLAB model is compiled using the MATLAB Compiler Toolbox, and it is hosted as a web application using Microsoft Internet Information Services (IIS) (The MathWorks Inc., 2022). We opted for this approach because it meant similar HTTP requests for both models.

3.3 Separation Simulation Model

The solution models' second part consists of separation processes that are simulated using commercial simulation software. The simulator model consists of three separator columns and estimates the separation, and thus it creates DHP unit's four different end products. Results from the first model are fed to the second model as significant inputs and the remaining operation parameters that the model requires are pulled with a similar simple SOAP call as explained previously. Because of this, 22 different features are pulled from the historian on top of the 145 features coming from the first model. A Python script utilizing the simulator's COM interface existing on its backend was used to feed these features to the simulator. The script consists of 4 different sections and employs the Win32com library to manipulate simulator classes and objects.

Figure 2: Live dashboard of the solution.

The application runs with the simulator already opened in the operating system, so the first section does not open the simulator but only attaches to the respective process, finding the flowsheet of the simulator and assigning individual stream and equipment to respective Python variables. Then, in the second section, the output of the first model, which is the input of the second model, is characterized as an oil mixture using the 145 features mentioned earlier. After that, it is attached to the input stream in the simulation flowsheet. The paused simulation is run to steady state in Section 3, and any errors are caught and handled in this section. If the simulation breaks due to any reason, such as bad data or a momentary server downtime, the simulation file is re-launched from a safe point for the next run in 15 minutes. And finally, in Section 4, the end results of the separator columns (temperatures and pressures mainly) and the product specifications that are significant for the daily operation are read from the simulator and fed back to Node-RED flow as a JSON file.

Though navigating the object-oriented topology of the simulator was challenging, the COM interface provided a crucial role in automating the software and streamlining task management. By automating this part of the solution, we were able to eliminate the most repetitive manual section. In addition, since the Python script manipulates the simulation and runs it to a steady state, we can read any important results from the simulator and send them back to the Node-RED flow as a JSON file. This operation takes less than 30 seconds, and the same simulator file can be used indefinitely if the simulation does not break. In which case, the simulation file is closed without being saved, reopened, and restarted from a safe point.

The simulator model is hosted as a web application using flask framework. Flask was chosen because it requires no additional tools and offers simplicity and flexibility through the implementation of a minimal web server. (Flask Documentation User's Guide, 2010).

3.4 Node-RED Flow

Node-RED is an open-source, versatile, flow-based development ETL tool (Rymaszewska et. al., 2017). JavaScript based interface, modifiable and flexible nature, low overhead are why Node-RED was used in this study. The explained communications in previous sections, besides the analyzer data communication, all occur in the respective Node-RED flow by the help of different processes.

Analyzer data is directly written to a specified directory in the Node-RED server by a bat script operating on its own computer. The latest written csv file in the analyzer directory is read by a node in the flow. SOAP calls that bring the process data to the flow is handled within a simple Python process for easier code maintenance. Both are combined to generate the total input data.

The Node-RED flow uses the input data in HTML request nodes to send requests to the models that are hosted at a local IP address as flask applications. Results are sent back to the Node-RED flow as JSONs. The direct estimated results that have significant and urgent operational meaning are fed to a simple Node-RED dashboard to be used as a decision support measure. The complete flow is on a loop that repeats itself every five minutes, thus the dashboard refreshes itself every five minutes. Four product specifications can be seen in color coded gauge graphs showing the operation engineers the estimated outcome of the current state (Fig. 2.).

The dashboard is hosted at a specific IP address and port that can be accessed by the authorized users on the company intranet. With this method, we aimed to let users see the live results directly. The entire results are written to a SQL database for further use and additional statistical analysis applications.

4 RESULTS AND DISCUSSION

This study demonstrated that we achieved our objective of creating a system that helps the operation make accurate decisions before any operational errors happen. Operating at 5-minute frequencies, the solution works with the previous 5-minute average unit data that has been collected from temperature, flow, and pressure indicators. The frequency was chosen as 5 minutes because the unit's operating procedures and operational parameters cannot change significantly faster than 5 minutes. Also, the overall flow of the solution (Fig.2) runs for 2 minutes before writing its results, so it could not run faster than that. However, 5 minutes was a balanced midpoint, as running the model more frequently would not provide any benefit.

Using live data streams in combination with predictive modelling can greatly improve the effectiveness of operations too. The properties of the feed, which include but are not limited to True Boiling Point, are fully utilized through the decision support tool to correct possible deviations in process variation before they occur. This is in great contrast to the backward mechanisms of feedback that mostly respond too late to avoid losses. An easy-to-use dashboard allows plant operators to make fast, informed decisions, reducing out-of-specification products and unnecessary reactor loadings.

Strong data management in the system is due to flow development by Node-RED and proper, secure data movement structure by SOAP APIs. The intuitive interface of Node-RED made the co-ordination possible with varying sources of data to run complex decision-making automated smoothly and efficiently. The choice of technologies, such as Node-RED, is based on the specific operational requirements of the organization technology structure, ensuring system efficiency and reliability.

Collecting all available data further enabled statistical modelling, data analysis, machine learning, and physical modelling work. One can exploit similar ease-of-access to live data to transfer their offline practices to an online context. In addition, by connecting the physical models we have developed to live data and automating them, we have enabled further simulations or mathematical models to work with live data with the methods we have developed in-house automatically. In addition, by linking our physical models to live data and automating them, we have facilitated the use of additional simulations or mathematical models with live data automatically using our proprietary methods. Given that simulations and models integrated with live data or databases are often sold commercially as separate packages or licences, this capability represents a significant economic advantage of our approach. Several issues arose, mainly, how to ensure that models could run faster than the decision support requirements and how complex data flows could be managed securely. For instance, an online analyzer was installed to model the chemical properties of the feed coming in, and an intermediate server was also installed for the secure handling of the data. Such is the kind of careful planning that has gone into building a balance between real-time processing capacity and data surety (Aldoseri et. al.,2023).

Economically, massive savings could be realized through real-time optimization of DHP unit operations by minimizing off-spec diesel and extending catalyst life (Aydin, 2015). Even hydrogen consumption is lowered under optimal conditions in the reactor, producing further decreases in operational costs. Environmentally, more controlled sulfur removal processes yield diesel products that meet and surpass-stringent environmental regulations, thereby limiting harmful emissions and producing greater sustainability.

Forthcoming, future research efforts might focus on refining the models to further enhance accuracy and improve the response times to higher levels. Predicting long-term trends in addition to predicting trends of potential issues would be a big added value toward the decision support. Further enhancement of the system to include interaction with other units within the refinery could provide a more comprehensive approach toward the refinery optimization by extending the benefits realized in the DHP unit across the facility.

5 CONCLUSIONS

In this paper, we tried to describe the automated application we developed by combining multiple different software and data sources to improve and support the current operation and reduce potential errors by giving them the ability to react before errors occur. We developed a data connection to two different data sources through the unit firewall to our server using SOAP calls and a simple bat script to access the data. By feeding the pre-processed versions of this data to the two models we developed in MATLAB and using commercial process simulators, we produced results to predict the course of the current operation. The complete connection between data points, and models and databases are done via the open-source project, Node-RED and we automated commercial simulators using the COM interface of the Windows operating system in Python and delivered live results to users in Node-RED interfaces.

In summary, with this decision support system, unit engineers will be able to make more controlled interventions, intervene with prior knowledge of product characteristics, operate in a manner that is more aligned with maintenance schedules, and follow production planning objectives.

REFERENCES

Aldoseri, A., Al-Khalifa, K. N., & Hamouda, A. M. (2023). Re-Thinking Data Strategy and Integration for Artificial intelligence: Concepts, opportunities, and challenges. Applied Sciences, 13(12), 7082. https://doi.org/10.3390/app13127082

Aydın, E., et al. “Dynamic modeling of an industrial diesel hydroprocessing plant by the method of continuous lumping,” Computers & Chemical Engineering, vol. 82, pp. 44–54, 2015, doi: 10.1016/j.compchemeng.2015.06.005.

Aydin, Erdal. (2015). Plant-Wide Modeling, Optimization and Control of an Industrial Diesel Hydroprocessing Plant. 10.13140/RG.2.2.23698.86725.

A. Gruia, Hydrotreating, in: D.J.S. Jones, P. Pujadó (Eds.) Handbook of Petroleum Processing, Springer, Netherlands, 2006, pp. 330-347.

A. Rymaszewska, P. Helo, and A. Gunasekaran, “IoT powered servitization of manufacturing – an exploratory case study,” International Journal of Production Economics, vol. 192, pp. 92–105, 2017, doi: 10.1016/j.ijpe.2017.02.016.

A. Safari and M. Vesali-Naseh, “Design and optimization of hydrodesulfurization process for liquefied petroleum gases,” Journal of Cleaner Production, vol. 220, pp. 1255–1264, 2019, doi: 10.1016/j.jclepro.2019.02.226.

A. Yasmal, G. K. Kaya, E. Oktay, C. Çölmekci, and E. Uzunlar, "Data Driven Leak Detection in a Real Heat Exchanger in an Oil Refinery," in Computer Aided Chemical Engineering, vol. 52, Elsevier, 2023, pp. 3091–3096. doi: 10.1016/B978-0-443-15274-0.50493-5.

F. S. Falla, C. Larini, G. A. C. Le Roux, F. H. Quina, L. F. L. Moro, and C. A. O. Nascimento, "Characterization of crude petroleum by NIR," Journal of Petroleum Science and Engineering, vol. 51, no. 1–2, pp. 127–137, 2006, doi: 10.1016/j.petrol.2005.11.014.

Flask Documentation User's Guide (2024) Flask Documentation (3.0.x). Available at: https://flask.palletsprojects.com/ (Accessed: 24 June 2024).

G. K. Kaya, P. Döloğlu, Ç. O. Özer, O. Şahin, A. Palazoğlu, and M. Külahçı, "A Study of Spectral Envelope Method for Multi-Cause Diagnosis using Industrial Data," in Computer Aided Chemical Engineering, vol. 50, Elsevier, 2021, pp. 1331–1337. doi: 10.1016/B978-0-323-88506-5.50205-9.

Halili, F., & Ramadani, E. (2018c). Web services: A comparison of soap and rest services. Modern Applied Science, 12(3), 175. https://doi.org/10.5539/mas.v12n3p175

H.-B. Park, K.-D. Kim, and Y.-K. Lee, "Promoting asphaltene conversion by tetralin for hydrocracking of petroleum pitch," Fuel, vol. 222, pp. 105–113, 2018, doi: 10.1016/j.fuel.2018.02.154.

J. W. Ward, "Hydrocracking processes and catalysts," Fuel Processing Technology, vol. 35, no. 1–2, pp. 55–85, 1993, doi: 10.1016/0378-3820(93)90085-I.

M. S. Rana, V. Sámano, J. Ancheyta, and J. A. I. Diaz, "A review of recent advances on process technologies for upgrading of heavy oils and residua," Fuel, vol. 86, no. 9, pp. 1216–1231, 2007, doi: 10.1016/j.fuel.2006.08.004.

"Automotive fuels. Unleaded petrol. Requirements and Test Methods," knowledge.bsigroup.com, Apr. 30, 2023. https://knowledge.bsigroup.com/products/automotive-fuels-unleaded-petrol-requirements-and-test-methods-5?version=standard&tab=preview

R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, Inf. Comput. Sci., Univ. California, Irvine, CA, USA, 2000.

T. Kabe, A. Ishihara, and W. Qian, "Hydrodesulfurization and hydrodenitrogenation. Chemistry and engineering," Jul. 1999, Accessed: Oct. 19, 2023. [Online]. Available: https://www.osti.gov/etdeweb/biblio/20101494

The MathWorks, Inc. (2022). MATLAB version: 9.13.0 (R2022b). Accessed: January 01, 2023. Available: https://www.mathworks.com

Word Wide Web Consortium, SOAP Version 1.2, W3C recommendation, second ed., http://www.w3.org/TR/soap, Feb. 2010.

APPENDIX

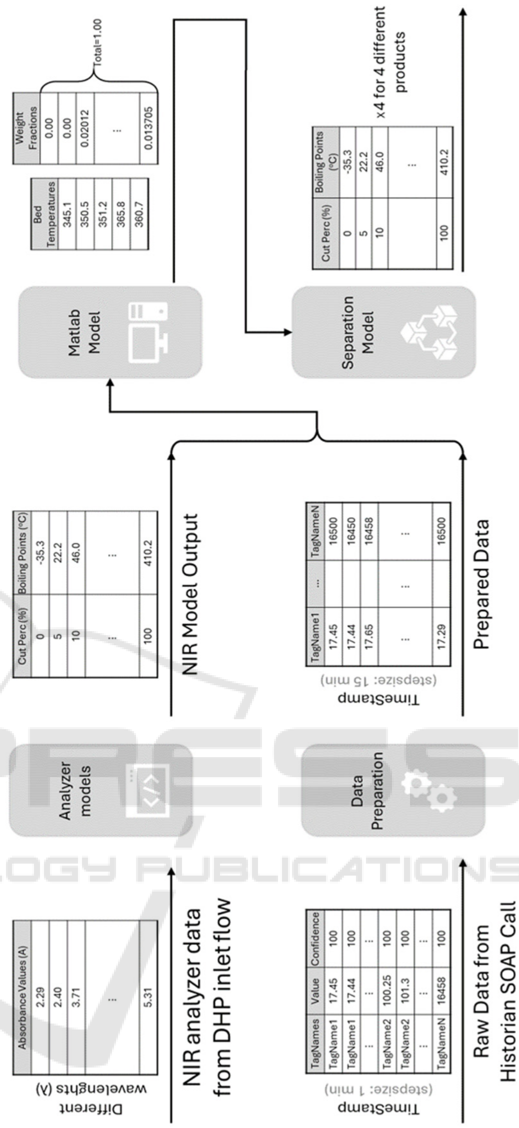

Figure 3.