Quantum Computing Application for Mapping Outputs of an Aspen-Python-Activity Browser interface, assisted by Support Vector Machines

Ada Josefina Robinsona\*, Viorica Sirghiia, Stavros Papadokonstantakisa

a Institute of Chemical, Environmental and Bioscience Engineering

TU Wien,1060 Wien, Austria

ada.robinson@tuwien.ac.at

Abstract

This paper investigates the advancements in Process System Engineering (PSE) by integrating computational methodologies and tools to incorporate next-generation technologies such as Support Vector Machine (SVM) metamodels and Quantum Computing into PSE workflows. We use Python programming language to create an interface that interconnects Aspen Plus and Activity Browser, a graphical user interface for the Brightway2 LCA framework, to accelerate process modelling, simulation, and Life Cycle Assessment (LCA) while bridging the gap between process simulation and environmental impact assessment.

We conduct multiple sensitivity analyses and use the automated interface framework to generate preliminary ReCiPE indicators for LCA. Additionally, we compare the performance of classical Support Vector Regression (SVR) models versus quantum SVR models. We transform classical machine learning models into quantum models using parametrized quantum circuits in Python's scikit-learn and Qiskit packages.

Our preliminary results demonstrate the quantum SVR capabilities to reinforce more efficient, accurate, and sustainable automated process simulation optimization for next-generation process design and assessment approaches.

Keywords: Quantum Computing, Life Cycle Assessment, Support Vector Machine, Machine Learning, Process Simulation

* 1. Introduction

Process Systems Engineering has become an indispensable and well-established tool for developing, designing, and optimizing chemical processes since its beginnings in the 1970s. Computational methodologies and tools made it feasible to model and simulate complex industrial operations, including traditional chemical processes as well as sustainable ones, enhanced by process intensification. Process simulations are used in the chemical industry to support the entire life cycle of a chemical process, including development, design, construction and operational optimization. Furthermore, simulations enable a holistic understanding of the environmental impact associated with a product or process by modelling and analysing the environmental effects from raw material extraction to disposal. However, when developing innovative processes, there is currently no single process simulation environment capable of adequately representing all aspects of a process's life cycle. Life Cycle Assessment (LCA) studies and Techno-Economic Analyses often result in numerous trial-and-error phases during technology upscaling, significantly extending time-to-market and costs (Algren et al., 2021).

To effectively integrate process simulation with other applications, such as conducting Life Cycle Assessment (LCA), is a challenging and resource-intensive task due to the way commercial software is integrated into a company's infrastructure. Furthermore, the advancing digitalization within industrial environments demands open interfaces, modularization, and efficient data connections for feasible process modeling, simulation, and optimization. Machine learning and data-driven models are now available and can adapt flexibly to large datasets to analyze the behavior of developing technology at an industrial scale and create scenarios for optimizing environmental factors while meeting higher-level application demands for accuracy, convergence, and speed.

To address the challenge of integrating and switching between these components based on specific application requirements, the utilization of Python can be a potent mean to accelerate process modelling, simulation, and Life Cycle Assessment (LCA) generation through an automated interface (Casas et al., 2020). The Activity Browser is an open-source software for LCA projects within the Brightway framework. It enables Python integration for LCA parameterization, scenario modeling, graph exploration, and other advanced features. This approach effectively bridges the gap that typically exists between process simulation and the assessment of environmental impacts in a streamlined and efficient manner,

Incorporating metamodeling strategies into complex process operations allows us to reduce model complexity and transform it into more manageable algebraic models. One effective way to achieve high accuracy and computational efficiency is by utilizing a Support Vector Machine (SVM) metamodel (Müller et al., 2017). However, the landscape of computational science is rapidly evolving, and a new horizon has emerged with the emergence of Quantum Computing. This presents a unique opportunity to seamlessly integrate quantum machine learning algorithms into the broader framework of metamodeling, optimization and decision-making. Implementing quantum machine learning algorithms could lead to faster training, improved data point separation, and better generalization performance, challenging traditional data science methods. We have decided to investigate the difference in performance between a classical (SVR) and a quantum (QSVR) support vector regression model, using the output data derived from an Aspen – Python – Activity Browser interface designed for simulating a hydrocarbon separation process, serving as an illustrative case study.

For the SVR model we used Python's *scikit-learn* library and for the quantum part *Qiskit* packages for quantum machine learning have been applied (Benedetti et al., 2019).

* 1. Methods
		1. Interface Integration

The Aspen-Python-Activity Browser framework is designed in a Jupyter Notebook to ensure flexibility and scalability in its functionality. This framework is the foundation for integrating various software tools and conducting comprehensive analyses (see fig. 2). The simulation section includes an open-source code library that facilitates the integration of simulations developed in Aspen Plus (ten Hagenet al., 2022). In our case study, we focus on the separation and conversion of a hydrocarbon mixture, which involves feed preparation, a DSTWU column, heat exchangers, and an RCSTR and a PLUG reactors to obtain main products (Prod1) and by-products, as shown in a simplified graphical representation in fig. 1. The simulation's inputs and outputs can be defined in this section. Upon linking the simulation, the outputs can be used as inputs for the environmental assessment section, where the process's life cycle assessment (LCA) is modelled upon Activity Browser, a graphical user interface for the Brightway2 open-source LCA framework (Steubing et al., 2020). The environmental section extracts LCA calculations and conducts analyses based on specific environmental metrics and data settings. We consider GWP (kg-CO2-eq/t prod1) and ReCiPe 2016 v1.03 endpoint (E) indicators, utilizing the ecoinvent 3.5 database (Köck et al., 2023).



Figure Simplified Process Flowchart for the Separation and Conversion of a Hydrocarbon Mixture comprising Propane, N-Butane, 1-Butene, Ethane, Propylene, and N-Pentane via Distillation (DSTWU) and Conversion (RCSTR and RPLUG) Units, Resulting in Two Distinct Product Streams

* + 1. Support Vector Regression

The main goal of regression analysis is to identify the relationship between dependent and independent parameters, in order to predict the target variable. SVR allows for significant improvements in the area of Process Systems Engineering by predicting parameters which play a major role in the decisional processes related to optimization (Pasetto et al., 2022). Predicting environmental impact indicators, such as GWP and ReCiPe, as well as specific operational parameters, like the light key rate within a DSTWU, can play a crucial role in the optimization of industrial processes from an environmental standpoint. This predictive approach aids in the identification of trade-offs among various environmental impact categories and their relationship to the desired product quality and yield. SVR can be applied for linear and non-linear regression problems, by implementing a kernel function such as sigmoid, polynomial or radial basis function (as in our case).

* + 1. Quantum Support Vector Regression

The main goal of QSVR is to enhance the performance of classical machine learning regression models, up to the point of outperforming classical methods. The QSVR model developed in our work is based on the quantum kernel method applied on a 2 qubits parametrized quantum circuit (Pasetto et al., 2022).



Figure Jupyter Notebook environment workflow for Python integration and data analysis, with ASPEN Plus for process simulation, Activity Browser along with the ecoInvent database for comprehensive environmental impact assessment, and optimization using Support Vector Regression (SVR) and Quantum Support Vector Regression (QSVR).

* 1. Results
		1. Dataset

The output of the LCA scenario resulted from the Python-Aspen-AB interface has served as the dataset for the modelling task. The training and test datasets consist of 3 variables as follows: Light Key (operation parameter of the DSTWU column) and GWP and ReCiPe, the first one was defined as the label and the next 2 as features (LCA environmental indicators), each with 141 datapoints. These variables were selected because of their strong correlation with the operational process performance under investigation. The light key component often serves as the primary product, and its separation can result in substantial energy consumption. Predicting the behaviour of the light key component can contribute to energy optimization, cost reduction, and mitigating environmental impacts. Furthermore, while GWP helps to identify conditions linked to high global warming potential, ReCiPe can aid in assessing trade-offs among various environmental impact categories when optimizing product quality and yield. For encoding classical data into quantum space, we have used a second-order Pauli-Z evolution circuit, named as ZZFeatureMap in Qiskit circuit library (Daspal, 2022; Qiskit, 2023). The most efficient model parameters such as data splitting subsets, hyperparameters for both SVR and QSVR, and quantum kernel parameters have been determined empirically. The performance of created models is illustrated in the following subchapters.

* + - 1. Classical SVR

As a first step, the performance of the classical model has been evaluated by comparing test and training accuracies, the coefficient of determination (R2), and the root mean square error (RMSE) for each of the predicted variables: GWP and ReCiPe, presented in table 1 (Scikit, 2023).

As expected, the classical SVR model performs very well, with a 100% accuracy and minor errors.

Table 1: R2 and RSME for the classical SVR model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Accuracy (train) | Accuracy (test) | R2 | RMSE |
| GWP | 1,00 | 1,00 | 1,00 | 0,15 |
| ReCiPe | 1,00 | 0,99 | 0,99 | 0,07 |

Figure 1 illustrates the parity plots between actual and predicted values and as it can be clearly seen, the predicted datapoints are nearly equal to the actual values.



Figure 1: Actual vs predicted target values for the classical SVR model applied on the test subset

* + - 1. Classical SVR vs Quantum SVR

Table 2 displays the comparison of models’ accuracy obtained for classical and quantum SVR, applied on both training (subset size = 0,7) and testing (subset size = 0.3) subsets. The performance of QSVR model with quantum kernels is almost equivalent to the classical SVR model. Nevertheless, the current status of QSVR packages proved to be more suitable for the classification problem, rather than the regression problem, therefore only the accuracy has been evaluated at this stage.

Table 2: Models’ accuracy for SVR and QSVR

|  |  |  |
| --- | --- | --- |
|  | Accuracy (train) | Accuracy (test) |
| SVR | 0,91 | 0,89 |
| QSVR | 0,89 | 0,90 |

The main reason for different accuracy outcomes for the SVR models in Table 1 and Table 2 is due to the fact that different values for random seeds have been tested. The main role of a random seed is to ensure the reproducibility of results and the fact that the code will give the same output after re-running it. As it can be seen, a variation in the initialization parameter has a dominant impact on the classical SVR model performance. At the same time the quantum SVR model does not seem to be affected by changing the random seed value, as in our current QSVR model the random state has been set to qiskit random number generator available in the algorithm utils global package (Qiskit, 2023).

* 1. Conclusions

The convergence of classical and quantum computing holds immense potential for revolutionizing optimization research across diverse domains. In our work, the performance of two regression models has been compared, one for a classical support vector regression and one for its quantum counterpart. Even though the QSVR does not present a significantly higher performance as compared to classical SVR, the potential of quantum machine learning models should be emphasized as it unlocks the way to exploring its potential for future applications on more complex problems and larger datasets. Furthermore, our ongoing work is dedicated to addressing the challenges of real-time data evaluation and dynamic simulations. Our aim is to facilitate the seamless integration of quantum-enhanced techniques into practical applications in process control, techno-economic analysis, and environmental optimization. This marks a holistic advancement in Process System Engineering and environmental sustainability

References

Algren, M., Fisher, W., Landis, A.E., 2021. Machine learning in life cycle assessment, in: Data Science Applied to Sustainability Analysis. Elsevier, pp. 167–190. https://doi.org/10.1016/B978-0-12-817976-5.00009-7

Benedetti, M., Lloyd, E., Sack, S., Fiorentini, M., 2019. Parameterized quantum circuits as machine learning models. Quantum Sci. Technol. 4, 043001. https://doi.org/10.1088/2058-9565/ab4eb5

Casas, M.G., Andía, J.D., Martos, J.L.G., 2020. ECO2DES: Python Framework for the Eco-Design of Industrial Processes, in: Computer Aided Chemical Engineering. Elsevier, pp. 1987–1992. https://doi.org/10.1016/B978-0-12-823377-1.50332-3

Daspal, A., 2022. OptiPauli: An algorithm to find a near-optimal Pauli Feature Map for Quantum Support Vector Classifiers, in: 2022 IEEE International Conference on Quantum Computing and Engineering (QCE). Presented at the 2022 IEEE International Conference on Quantum Computing and Engineering (QCE), IEEE, Broomfield, CO, USA, pp. 828–830. https://doi.org/10.1109/QCE53715.2022.00133

Köck, B., Friedl, A., Serna Loaiza, S., Wukovits, W., Mihalyi-Schneider, B., 2023. Automation of Life Cycle Assessment—A Critical Review of Developments in the Field of Life Cycle Inventory Analysis. Sustainability 15, 5531. https://doi.org/10.3390/su15065531

Müller, A.C., 2016, Introduction to Machine Learning with Python.

Pasetto, E., Delilbasic, A., Cavallaro, G., Willsch, M., Melgani, F., Riedel, M., Michielsen, K., 2022. Quantum Support Vector Regression for Biophysical Variable Estimation in Remote Sensing, in: IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium. Presented at the IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium, IEEE, Kuala Lumpur, Malaysia, pp. 4903–4906. https://doi.org/10.1109/IGARSS46834.2022.9883963

Qiskit open-source SDK. https://qiskit.org/

Scikit learn open-source. Metrics and scoring: quantifying the quality of predictions. https://scikit-learn.org/stable/modules/model\_evaluation.html#regression-metrics

Steubing, B., De Koning, D., Haas, A., Mutel, C.L., 2020. The Activity Browser — An open source LCA software building on top of the brightway framework. Software Impacts 3, 100012. https://doi.org/10.1016/j.simpa.2019.100012

ten Hagen, Richard Wolfgang, 2022, AspenPlus Python Interface (Version 2.1), <https://github.com/YouMayCallMeJesus/AspenPlus-Python-Interface/tree/main>.