Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 225 (2023) 4560–4569

Procedia Computer Science

www.elsevier.com/locate/procedia

27th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2023)

A Web-Based Platform for Efficient and Robust Simulation of Aquaculture Systems using Integrated Intelligent Agents

Aya Saad[a,*], Biao Su[a], Finn Olav Bjørnson[a]

[a]*Aquaculture Department, Sintef Ocean AS, Trondheim, Norway*

Abstract

We propose a web-based platform of integrated intelligent agents that incorporates multiple Functional Mock-up Units (FMUs) and surrogate modelling techniques. In this platform, each FMU envelops a stand-alone simulation component that represents an aquaculture system, such as the fish growth model, water quality model, and fish behavior model. Some FMUs may be computationally expensive to simulate or have different time step intervals, making integration with other FMUs difficult. To address these challenges, we employed surrogate models to substitute the more computationally expensive models. In this work, surrogate models are trained using simulation data and selected based on robustness analysis to ensure the overall system input-output reliability. The platform also includes a Chatbot component utilizing natural language processing and decision-making techniques to interpret user requests and provide tailored FMU configurations, enhancing the user simulation experience. Overall, the proposed platform provides a comprehensive and efficient approach to modelling and simulating complex systems such as fish farms. Robustness analysis ensures the platform's accuracy and reliability, while the user-friendly interface enables easy tailored experimentation. By providing a framework for exploring the potential of aquaculture as a key source of food and income, the proposed platform represents a valuable interactive simulation tool for researchers, policymakers, and industry professionals seeking to improve the sustainability, efficiency, and economic viability of the aquaculture industry.

Keywords: Simulation; FMU; FMI; Surrogate models; Robustness analysis; Robust AI; Chatbot, BERT

1. Introduction

Simulation has long been an essential tool for researchers, policymakers, and industry professionals to better understand complex systems and explore potential solutions. In recent years, there has been a growing interest in simulation-based approaches to address challenges in aquaculture, particularly as the global demand for seafood continues to increase [5, 12, 20, 1, 23]. Aquaculture is an important source of food and income for many communities

* Corresponding author.
E-mail address: aya.saad@sintef.no

10.1016/j.procs.2023.10.454

worldwide, but it faces significant challenges related to water quality, disease outbreaks, and environmental sustainability. Efforts to model and simulate aquaculture systems have faced challenges due to the computational complexity of the models involved. Specifically, models that involve biological processes can be computationally expensive and limited in their ability to represent the complex interactions between biological and physical components [11, 2, 25]. Integrating models with different time step intervals can also be challenging, further complicating the task of simulating aquaculture systems. Despite these challenges, the need for accurate and efficient modelling of aquaculture systems has spurred ongoing research and development in this area.

To address the challenges of modelling and simulating aquaculture systems, we propose the design and implementation of a web-based platform of a system of integrated intelligent agents. The contribution of this paper is fourfolds:

- Design and implement an online system that integrates simulations of various components essential for creating digital twins of different aquaculture systems. By incorporating robustness analysis, the system ensures the effective functioning and reliable performance of the integrated components across diverse circumstances.
- Employ surrogate modelling to substitute computationally complex components that are difficult to integrate, thereby facilitating efficient simulations. The surrogate models, developed and validated through robustness analysis, provide faster integration without compromising accuracy. This approach significantly enhances the platform's performance by reducing computational time and effort.
- Deploy natural language processing techniques to allow end-users to customize and tailor their simulation setups easily. The incorporation of robustness analysis in the natural language processing capabilities ensures that the system accurately interprets and responds to user inputs, enhancing the overall user experience and minimizing the risk of errors in simulation configurations.
- Emphasize the benefits of our system to end-users, enabling them to make informed decisions. Through robustness analysis, the system continually evaluates and improves its reasoning and execution processes, resulting in reliable and trustworthy simulation outcomes.

By integrating robustness analysis across these contributions, the web-based platform not only provides advanced simulation capabilities but also ensures the system's accuracy and adaptability in addressing the challenges of aquaculture system modelling.

The proposed platform includes several components that work together to create a seamless, tailored and efficient simulation experience. The first component is the integration of multiple Functional Mock-up Units (FMUs) [6], which incorporate and represent different components of the aquaculture system, such as the fish growth model [24], the water quality model [27], and the fish behavior model [25]. Some of these FMUs may be computationally expensive to simulate or may have different time step intervals, making them difficult to integrate with other FMUs. To overcome these issues, we use surrogate modelling techniques [7] to create surrogate models that act as substitutes for the more computationally expensive models. These surrogate models are trained using simulation data and carefully chosen based on robustness analysis [15, 22] to ensure they are accurate and reliable. The use of surrogate modelling techniques reduces the computational cost and increases the efficiency of the simulations, especially when involving integrated FMUs. In addition, the platform incorporates a Chatbot component based on fine-tuning the well-known Transformer model [28, 26, 17]. This Chatbot is trained to assist users in configuring simulations of different experiments based on their input. The Chatbot Assistant utilizes natural language processing and decision-making techniques to interpret the user's requests and provide a tailored list of FMU configurations to enhance their simulation experiences. The incorporation of the Chatbot component provides an easy-to-use and flexible interface that allows users to interact with the system and perform simulations with ease.

Overall, the integration of multiple FMUs and the use of surrogate modelling and Chatbot components offer a comprehensive and efficient approach to modelling and simulating complex systems such as aquaculture. The proposed platform provides a user-friendly interface that enables users to simulate different experiments based on their input, and its accuracy and reliability are ensured through robustness analysis. As such, it represents a valuable tool for researchers, policymakers, and industry professionals seeking to improve the sustainability, efficiency, and economic viability of the aquaculture industry.

In summary, the development of simulation-based approaches for aquaculture systems is crucial to meeting the growing global demand for seafood while addressing the environmental and sustainability challenges facing the industry. The proposed web-based platform offers an innovative solution to the computational complexity of modelling interconnected FMUs, while also providing a flexible and efficient interface for users. By integrating surrogate modelling and Chatbot components, the platform enhances the accuracy, reliability, and ease of use of aquaculture simulations, making it an ideal tool for researchers, policymakers, and industry professionals. We believe that this platform has the potential to significantly contribute to the sustainability and growth of the aquaculture industry and provide a framework for exploring the potential of aquaculture as a key source of food and income.

2. Background

Rapid growth in computational technology over the last decades has stimulated an increased interest in using mathematical modelling as a tool to predict and replicate the dynamics in aquaculture production systems. Most of these models have specific focus domains and application areas as they are often developed on a case-by-case basis with specific applications or situations in mind.

Since not all models within the aquaculture domain are built on the same assumptions integration is difficult. Traditionally this has been solved through monolithic implementations [13, 25], where submodels are reintegrated and adapted to a specific case or setting. Recently, more interest has been given to co-simulation that has provided interesting results both in the automotive as well as the maritime domain. In co-simulation the focus is on creating independent models with loose coupling between subsystems [14], thereby reducing the time and effort needed to set up new simulations and different combinations of models.

A master algorithm is responsible for advancing each of the subsystems and making sure they share data at specific integration steps. As such co-simulation is ideally suited for what we are trying to achieve in this work. However, performance is still bound by the slowest running subsystem, and with simulations running at steps that are several thousands of orders of magnitude in difference, we needed to investigate the possibility of speeding up some parts of the process. This is where we started investigating the surrogate model [7] as a way of achieving faster integration without sacrificing too much accuracy.

Herein, the fish behavior model is selected for the investigation, as it is running at steps that are much smaller than the connected fish growth model, and it is intended to simulate the swimming activity and feed intake of each individual [11, 2] while the fish growth model only needs the average inputs for the entire population. The deformation of a flexible sea cage (e.g., due to water current) and its influence on the contained fish (up to 200,000 individuals) are also considered in the fish behavior model [25], therefore it is complex and well suited for surrogate modelling.

Additionally, we emphasize in this work the importance of robustness analysis to ensure the effective functioning of integrated systems and components in various circumstances. Robustness analysis involves evaluating the accuracy and dependability of a system's decision-making and execution processes through the creation of a range of hypotheses and models that validate decisions or actions, making it more resilient to random variations [15, 22]. The quality of input data and the manner in which actions are executed while engaging with the environment are crucial factors that influence system performance. Hence, robustness analysis is applied to ensure that data generated covers the entire variable domain and measure the system's performance under different situations. Incorporating feedback loops to observe and improve the system's behavior is also critical to robustness analysis.

To the best of our knowledge, there are limited efforts in the literature that address the challenges associated with integrating FMUs in a web-based platform that incorporates natural language processing and surrogate modelling techniques. This work aims to contribute to the development of a more efficient and integrated simulation tool for the aquaculture industry.

3. Methodology

The proposed framework enables virtual experiments as a digital twin to aquaculture research facilities, where the production systems (e.g., sea cages), environments (e.g., current, wave and water quality) and farmed fish (e.g., swimming activity and growth of Atlantic salmon) can be modelled considering the inputs from sensors, equipment and experimenters. The framework, as depicted in figure 1 consists of four main components: a web user interface (orange), a Bidirectional Encoder Representations (BERT) assistant (green), an XML predefined simulation configuration (purple), and a co-simulation platform that supports the FMI standard [6, 9] to seamlessly connect various

FMUs (blue). The web interface is available to the public [1] and linked with the technical framework integrating the models so that users can execute virtual experiments without needing a deep insight into how the framework and submodel integration are implemented. Regular users can run standardised simulations using the provided FMU's and xml templates. More advanced users have no restrictions on how they want to combine the submodels and can also include their own FMU's in the simulation should they wish to do so.

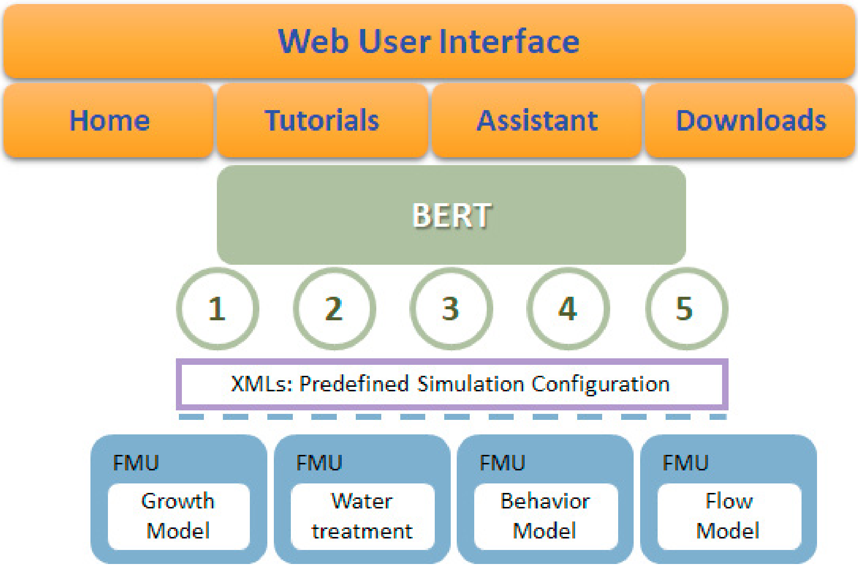

Fig. 1. The web user interface architecture

3.1. The Web User Interface (WUI) (orange)

The WUI is developed using Django [10] and it is designed to provide a user-friendly experience for a diverse range of users, from beginners to experts. It is compatible with various databases and employs Bootstrap, an open-source toolkit, to enhance the user experience. Bootstrap provides a comprehensive set of pre-designed and customizable components, including responsive layouts and interactive elements, enabling the WUI to deliver a visually appealing and intuitive interface. Additionally, the user interface features a simple and intuitive design, incorporating tutorials, an assistant interactive interface based on BERT, and quick access links for downloads. The interface also allows users to specify the infrastructure setup, biomass, and feeding regime, and suggests tailored experimental configurations.

3.2. The Assistant "BERT" (green)

The Assistant interface is linked to a language representation model BERT, which stands for Bidirectional Encoder Representations from Transformers [28, 26, 17]. The model is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. BERT is conceptually simple and empirically powerful and can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task-specific architecture modifications. The model deployed is trained over a 156060 number of phrases with 5 different classes, where each class represents a simulation setup for interconnected FMU based on the user input. The test accuracy is 0.82 over 100100 test phrases. Figure 2 shows the Assistant model user interface where users are prompted to input their free text describing a simulation scenario. The BERT model, in turn, interprets this user input and accordingly directs the user towards tailored simulation experiment configurations. It's worth noting that with more user input and feedback,

[1] https://ae2020virtuallab.sintef.no/VL3/

the BERT model performance can be further improved, achieving a measure of robustness analysis when users interact with the model over time.

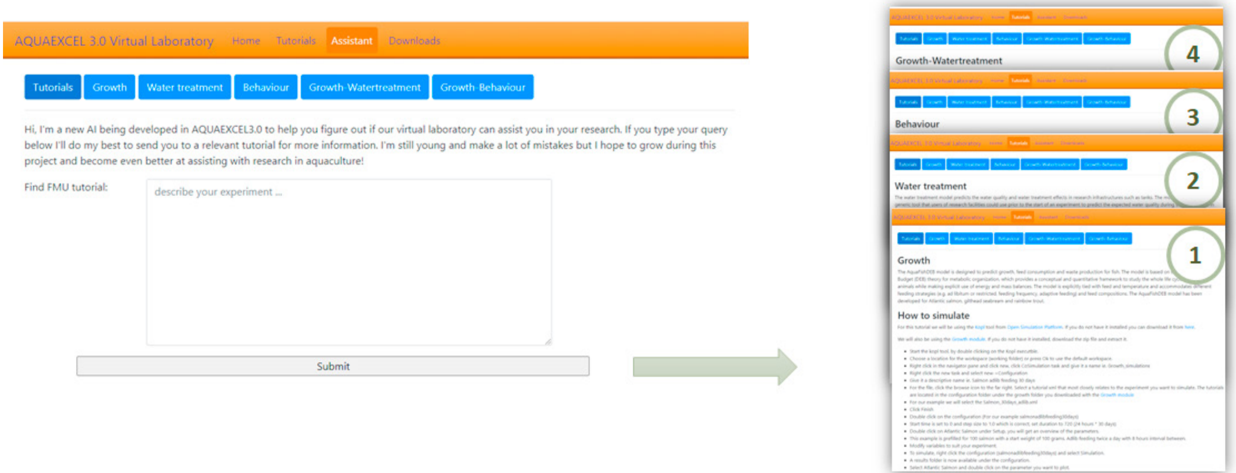

Fig. 2. The Chatbot in action: directing users to the experiment based on the input text

3.3. The XML Predefined Simulation Configuration (purple)

In this component, the input-output for the models is defined, and the interfaces between the FMUs are specified. The interfaces specify how the inputs and outputs of each FMU are connected to the inputs and outputs of other FMUs. The co-simulation platform provides an environment for simulating the FMUs together and exchanging data between them. After importing an XML template, the simulation can be further configured by setting the simulation time, step size, and other parameters.

3.4. The Functional Mock-up Units (FMU)s (blue)

FMUs are models that represent systems and components, and can be transferred between various simulation tools. These FMUs are employed in co-simulation, wherein each FMU can be used to simulate individual stand-alone systems. For example, the growth model FMU is built to envelop the differential equation-based growth model presented in [24], and the water treatment model FMU envelops the dynamic reactor model presented in [27]. The fish behavior FMU is built to envelop an agent-based model described in [25]. The integrated FMUs are built by connecting the growth model FMU with either the water treatment or fish behavior FMU. Each FMU is built as an independent software component that implements a set of input and output variables. These variables are used to transfer data between the FMUs during co-simulation. The variables can be either continuous (e.g. water temperature, fish weight) or discrete (e.g. feeding rate).

Overall, the web platform is built on the integration of various models using FMUs and the FMI standard, offering an intuitive front-end for co-simulation experiments. The BERT-based assistant further streamlines the simulation setup process, enabling users to customize their simulations accurately, focusing on their research and not the technical details of simulation setup and execution. Additionally, the platform's flexible design facilitates the expansion and integration of new models, allowing for the incorporation of additional features in the future. The interactive web interface created by integrating these four components enables researchers to simulate different aquaculture systems, explore the effects of various parameters, and build digital twin simulations without costly physical experiments.

4. Implementation

The web-based platform is built on top of Django [10], a high-level Python web framework that provides a robust and flexible environment for developing web applications. **Django** is based on the Model-View-Template (MVT) architecture, which separates the application's data model, user interface, and display logic. This separation enables developers to write clean, maintainable, and scalable code. Leveraging the many features of Django, the platform offers an intuitive and user-friendly web interface that allows users, regardless of their expertise level, to easily create tailored co-simulation experiments. The web interface provides a range of tools for creating and executing experiments, while the underlying framework ensures scalability and stability. The back-end SQL database is designed to store various details, such as simulation experiments, input-output variables of FMUs, XML configuration files, and data on model interactions and connectivity. This flexible design enables end-users and developers to seamlessly add more simulation experiments and integration, ensuring scalability in the long run.

To further optimize the user experience, the web-based platform incorporates a **BERT-based assistant** within the web user interface. This advanced feature employs the state-of-the-art natural language processing model, Bidirectional Encoder Representations from Transformers (BERT), to suggest appropriate simulation configurations based on user inputs. This feature ensures that users can create precise simulation setups while minimizing the risk of errors. The ”save” feature plays a crucial role in enhancing the robustness of the BERT model. By collecting user interactions and feedback, it incrementally expands the dataset, significantly increasing its diversity and coverage. This expanded dataset enables the BERT model to learn from a wide range of user inputs and effectively respond to various simulation scenarios. This iterative process of incorporating user feedback serves as a feedback loop, continuously improving the performance and reliability of the BERT-based assistant.

Currently, the platform has a dataset of (256160) samples spanning 5 different classes, each representing a distinct simulation setup. This dataset, combined with the robustness analysis measure facilitated by continually storing the user feedback on BERT responses provides greater flexibility and scalability for both end-users and developers.

In order to integrate the different **FMUs** into the co-simulation, we use the Functional Mock-up Interface (FMI) standard [6, 9]. The FMI standard provides a way to package FMUs and their dependencies into a single file, which can then be imported into the co-simulation platform. This makes it easy to exchange models between different simulation tools and to reuse models in different applications. During co-simulation, the different FMUs are executed simultaneously, and their inputs and outputs are exchanged according to the interfaces defined in the FMI file. The co-simulation platform ensures that the simulation time and step size are synchronized between the FMUs, so that they can communicate with each other at the correct time.

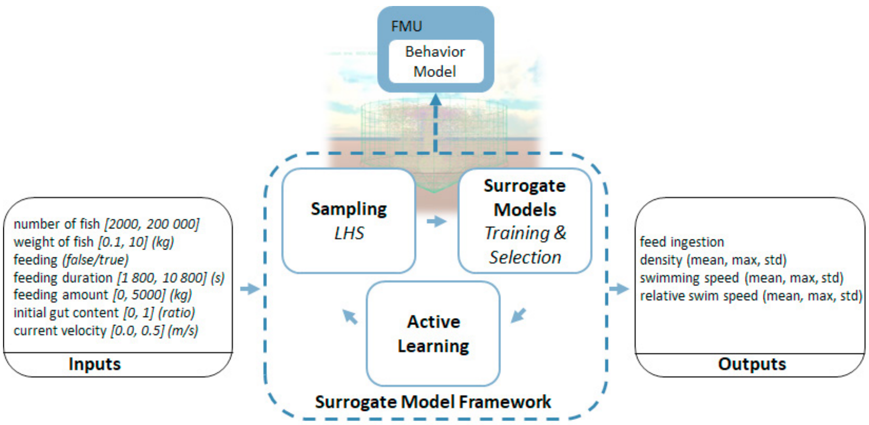

Fig. 3. Surrogate model workflow

The integration of FMUs into the co-simulation platform posed several challenges. One such challenge was the varying time step simulations across different FMUs, ranging from 0.1 second (e.g., the fish behavior model) to one hour (e.g., the fish growth model) and even days or months. Additionally, some FMUs were complex and slow, while others involved numerous variables and parameters, further complicating their integration into the overall simulation.

Surrogate modelling was adopted to overcome these complexities, whereby a simpler model was used to approximate the behavior of a more complex one [4, 29, 3, 21]. A proof of concept was demonstrated through the replacement of the Fish Behavior Model FMU with a surrogate model that accurately mimicked its behavior and was simpler to integrate and more efficient.

To build surrogate models for the fish behavior, a dataset with inputs matching those of the complex model is required. This dataset is generated by running the fish behavior model through an excessive number of simulations. To ensure that the input domain space is fairly covered, a Latin Hypercube Sampling approach [18] is employed generating 2000 different samples within the value limits of each input parameter. The inputs for the fish behavior model include the number of fish, with a range of 2000 to 200000, weight of fish with a range of 0.1 to 10 (*kg*), feeding a boolean value (true/false) indicating whether the fish are being fed at the moment, feeding duration with a range of 1800 to 10800 (sec), feeding amount with a range of 0 to 5000 (*kg*), initial gut content as a ratio of maximum gut volume between 0 and 1, and current velocity with a range of 0.0 to 0.5 (*m/s*). The model generates outputs that include feed ingestion, density (mean, maximum, and standard deviation), fish swimming speed (mean, maximum, and standard deviation), and fish swimming speed that is relative to the water current (mean, maximum, and standard deviation). As a result, the input dataset has a shape of (2000, 6) and the output dataset has a shape of (2000, 10).

Figure 3 illustrates the surrogate model selection workflow, which involves sampling, training, active learning, and testing. The Latin Hypercube **sampling** [18] scheme is adopted to generate initial training samples that are evenly distributed across the input parameter space. The model provides predictions at testing samples and estimates the associated prediction uncertainty, enabling **active learning** [8]. Refining the surrogate model is done iteratively, e.g. one sample is generated at a time, prioritizing the current model's expected prediction error.

Fourteen different surrogate models with their variations were tested and evaluated, including Support Vector Regression (SVR), Least Squares (LS), Quadratic Programming (QP), Multivariate Adaptive Regression Splines (MARS), Artificial Neural Network (ANN), Kriging, Random Forest (RF), Kernel Partial Least Squares (KPLS), Kernel Partial Least Squares with Kernels (KPLSK), Inverse Distance Weighting (IDW), Radial Basis Function (RBF), Gaussian Process Regression (GPR), and Radial Basis Function Network (RBFN). The detailed implementations of these models can be found in [3, 21, 16, 30, 19, 29, 4]. SVR, LS, QP, and MARS build predictive models, while ANN uses interconnected layers of nodes to produce an output. Kriging uses a hierarchical structure, RF is an ensemble learning method, and KPLS and KPLSK are useful for high-dimensional datasets and nonlinear relationships. IDW estimates a point's value based on its neighboring points, while RBF and GPR are useful for nonlinear relationships. RBFN is an artificial neural network, and ELM is a feedforward neural network.

5. Results and discussion

To evaluate the potential of surrogate models as replacements for complex fish behavior models, a five-fold cross-validation method was employed to determine the most suitable model with robustness analysis. The surrogate model with the best performance is selected based on its ability to accurately predict the target variable. Performance metrics, including R^2 score, Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Normalized Root Mean Squared Error (NRMSE), were calculated and averaged over the five folds of the cross-validation algorithm. R^2 score measures the proportion of variance in the output variable explained by the model's input variables. MSE calculates the average squared difference between predicted and actual values, RMSE is the square root of the average of the squared differences between the predicted and actual values, MAE measures the average absolute difference between predicted and actual values, and NRMSE is the normalized root mean square error between the predicted and actual output.

The results in Table 1 demonstrate that the Gaussian Process Regression (GPR) model performed the best among the surrogate models, as it has the highest R^2 score (0.99) and the lowest values for MSE, RMSE, MAE, and NRMSE. Both KPLS and its variation KPLSK models followed GPR in performance, while IDW was the worst performer.

In addition, the GPR model is highly efficient, as it took only 0.0241 milliseconds to execute, whereas the original fish behavior model required about 20 minutes per scenario (i.e., input sample). This significant reduction in simulation time makes GPR a highly attractive option for integration into co-simulations without adding additional overhead. These findings provide evidence that the surrogate GPR can effectively replace the fish behavior model in the co-simulation, producing similar simulation outputs but with significantly faster run times. This improvement can enhance the user experience by reducing the time required to obtain results, allowing researchers to focus on their experiments rather than waiting for simulations to complete.

Table 1. Surrogate models performance metrics

Model	R^2 score	MSE	RMSE	MAE	NRMS
SVR	0.917	0.003	0.055	0.045	0.118
LS	0.830	0.006	0.079	0.058	0.168
QP	0.955	0.001	0.040	0.028	0.087
MARS	0.955	0.001	0.040	0.028	0.087
KRIGING	0.988	0.0004	0.020	0.011	0.044
KPLS	**0.989**	**0.0004**	**0.019**	**0.011**	**0.042**
KPLSK	**0.989**	**0.0004**	**0.019**	**0.010**	**0.042**
IDW	0.412	0.023	0.152	0.119	0.324
RBF	0.964	0.001	0.037	0.018	0.078
GPR	**0.992**	**0.0003**	**0.017**	**0.010**	**0.037**
RF	0.958	0.001	0.039	0.025	0.084
RBFN	0.886	0.004	0.066	0.048	0.142
ANN	0.848	0.005	0.074	0.054	0.158
ELM	0.727	0.010	0.102	0.076	0.218

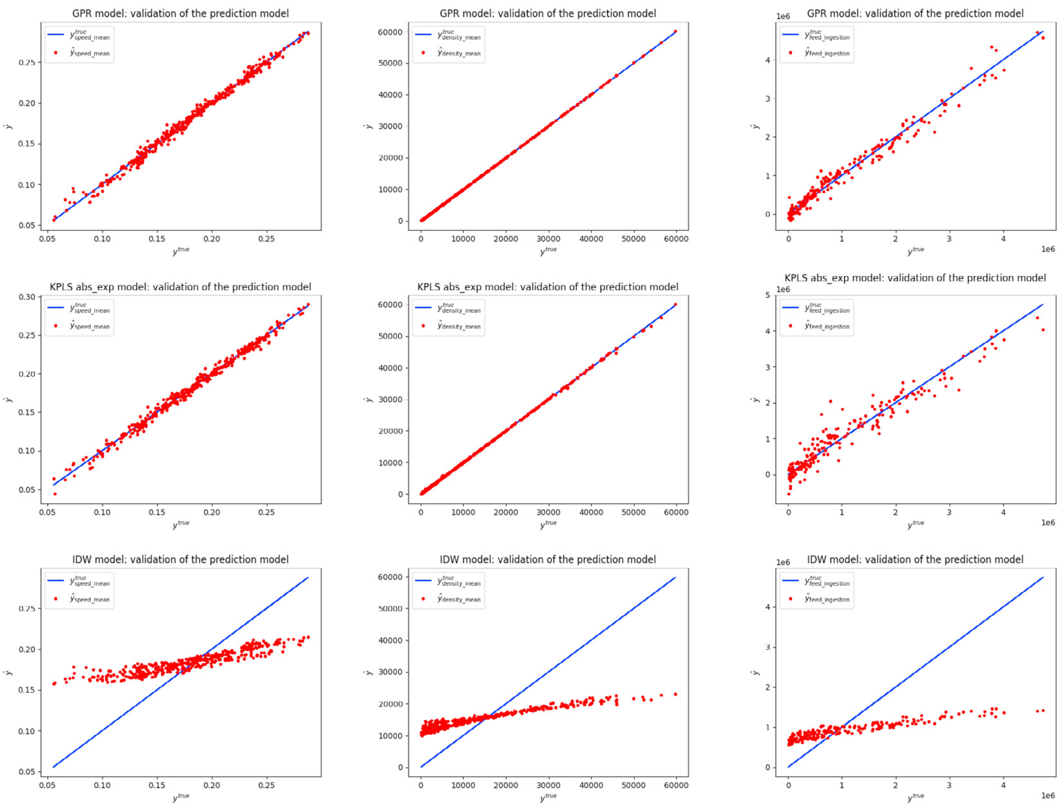

Fig. 4. The output ground truth y^{true} vs. model predictions $\hat{y}$. The top row is the GPR model predictions (highest performance R^2 score 0.99). The second row is the KPLS model predictions (second-best performer with R^2 score 0.98). The third row is the IDW model prediction (the lowest R^2 score 0.41).

Figure 4 illustrates the comparison between the output ground truth y^{true} and the model predictions $\hat{y}$ for three distinct output variables: fish mean swimming speed relative to the water current speed, fish mean density distribution along the cage, and feed ingestion or the average amount of food consumed by a fish in the cage. The first row displays the GPR model predictions, which achieved the highest R^2 score, the second row exhibits the QP model predictions, which ranked second in performance, while the third row showcases the kriging model predictions, which achieved the lowest R^2 score. These findings are consistent with the performance metrics listed in Table 1. The GPR model produced the least scattered predicted output, which corresponds more closely to the ground truth.

To ensure **robustness analysis** on the overall system, we implemented several measures. First, we applied robustness analysis on the BERT model by incorporating more user input and feedback during interactions, leading to an incremental build-up of the model's training dataset. This approach resulted in better interaction results, catering to user requests and continuously improving the BERT model's performance in the long run.

Second, we designed the web platform with a modular and flexible approach by separating the application's data model, user interface, and control logic. This separation of concerns makes it easy to integrate new models seamlessly, ensuring the platform's scalability for end-users and developers.

Third, we utilized the five-fold cross-validation method to determine the most suitable surrogate model for the overall system input-output reliability as a measure of robustness analysis. We also selected sample data based on Latin hypercube to ensure the dataset covers the variable domain as a whole.

Lastly, we replaced complex models, such as the fish behavior model, with faster models and enveloped them into FMUs as a measure of robustness analysis. This approach made it possible to seamlessly integrate new intelligent agents, performing specific simulations, to interact within the co-simulation platform, saving end-users time and effort and enhancing their user experience.

6. Conclusions and further work

In conclusion, the proposed web-based platform represents an innovative and comprehensive approach to modelling and simulating complex aquaculture systems. By integrating multiple FMUs and employing surrogate modelling techniques to substitute computationally expensive models, the platform ensures the accuracy and reliability of simulations while reducing computational cost and increasing efficiency. The incorporation of a Chatbot component based on natural language processing and decision-making techniques provides a user-friendly interface that allows users to easily interact with the system and configure simulations based on their input. The platform represents a valuable tool for researchers, policymakers, and industry professionals seeking to improve the sustainability, efficiency, and economic viability of the aquaculture industry.

In terms of future work, there are several opportunities to further enhance and expand the proposed platform. One potential avenue for improvement is the inclusion of more FMUs representing additional components of the aquaculture system, such as feed management systems and disease models. Furthermore, the Chatbot component could be expanded to incorporate more advanced natural language processing techniques and machine learning algorithms to improve its accuracy and responsiveness. Finally, the platform could be extended to include a wider range of aquaculture systems, enabling researchers and industry professionals to explore the potential of aquaculture as a key source of food and income in different contexts.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 871108 (AQUAEXCEL3.0). This output reflects only the author's view and the European Commission cannot be held responsible for any use that may be made of the information contained therein.

References

[1] Industry collaboration enables big data analytics, Mar 2022. URL: https://aquacloud.ai/about/.

[2] Morten O Alver, Kristoffer Rist Skøien, Martin Føre, Turid Synnøve Aas, Maike Oehme, and Jo Arve Alfredsen. Modelling of surface and 3d pellet distribution in atlantic salmon (salmo salar l.) cages. *Aquacultural engineering*, 72:20–29, 2016.

[3] Claudio Angione, Eric Silverman, and Elisabeth Yaneske. Using machine learning as a surrogate model for agent-based simulations. *PloS one*, 17(2):e0263150, 2022.

[4] Michael J Asher, Barry FW Croke, Anthony J Jakeman, and Luk JM Peeters. A review of surrogate models and their application to groundwater modeling. *Water Resources Research*, 51(8):5957–5973, 2015.

[5] Hans V Bjelland, Martin Føre, Pål Lader, David Kristiansen, Ingunn M Holmen, Arne Fredheim, Esten I Grøtli, Dariusz E Fathi, Frode Oppedal, Ingrid B Utne, et al. Exposed aquaculture in norway. In *OCEANS 2015-MTS/IEEE Washington*, pages 1–10. IEEE, 2015.

[6] Torsten Blochwitz. Functional mock-up interface for model exchange and co-simulation. *July [Online] Https://Www. Fmi-Standard. Org/Downloads (Accessed January 2016)*, 2014.

[7] Mohamed Amine Bouhlel, John T. Hwang, Nathalie Bartoli, Rémi Lafage, Joseph Morlier, and Joaquim R. R. A. Martins. A python surrogate modeling framework with derivatives. *Advances in Engineering Software*, page 102662, 2019. doi:https://doi.org/10.1016/j.advengsoft.2019.03.005.

[8] David A Cohn, Zoubin Ghahramani, and Michael I Jordan. Active learning with statistical models. *Journal of artificial intelligence research*, 4:129–145, 1996.

[9] Lukas Exel, Georg Frey, Gerrit Wolf, and Mathias Oppelt. Re-use of existing simulation models for dcs engineering via the functional mock-up interface. In *Proceedings of the 2014 IEEE Emerging Technology and Factory Automation (ETFA)*, pages 1–4. IEEE, 2014.

[10] Jeff Forcier, Paul Bissex, and Wesley J Chun. *Python web development with Django*. Addison-Wesley Professional, 2008.

[11] Martin Føre, Tim Dempster, Jo Arve Alfredsen, Vegar Johansen, and David Johansson. Modelling of atlantic salmon (salmo salar l.) behaviour in sea-cages: A lagrangian approach. *Aquaculture*, 288(3-4):196–204, 2009.

[12] Martin Føre, Kevin Frank, Tomas Norton, Eirik Svendsen, Jo Arve Alfredsen, Tim Dempster, Harkaitz Eguiraun, Win Watson, Annette Stahl, Leif Magne Sunde, et al. Precision fish farming: A new framework to improve production in aquaculture. *biosystems engineering*, 173:176–193, 2018.

[13] Martin Føre, Morten Omholt Alver, Jo Arve Alfredsen, Gunnar Senneset, Åsa Espmark, and Bendik Fyhn Terjesen. Modelling how the physical scale of experimental tanks affects salmon growth performance. *Aquaculture*, 495:731–737, October 2018.

[14] Cláudio Gomes, Casper Thule, David Broman, Peter Gorm Larsen, and Hans Vangheluwe. Co-simulation: A survey. *ACM Computing Surveys*, 51(3):49, 2018. doi:10.1145/3179993.

[15] Anne Håkansson, Aya Saad, Akhil Anand, Vilde Gjærum, Haakon Robinson, and Katrine Seel. Robust reasoning for autonomous cyber-physical systems in dynamic environments. *Procedia Computer Science*, 192:3966–3978, 2021.

[16] Sigve Karolius, Heinz A Preisig, and Henrik Rusche. Multi-scale modelling software framework facilitating simulation of interconnected scales using surrogate-models. In *Computer Aided Chemical Engineering*, volume 38, pages 463–468. Elsevier, 2016.

[17] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[18] Wei-Liem Loh. On latin hypercube sampling. *The annals of statistics*, 24(5):2058–2080, 1996.

[19] Ali Mehmani, Souma Chowdhury, and Achille Messac. Predictive quantification of surrogate model fidelity based on modal variations with sample density. *Structural and Multidisciplinary Optimization*, 52:353–373, 2015.

[20] Florian Perabo, Daeseong Park, Mehdi Karbalaye Zadeh, Øyvind Smogeli, and Levi Jamt. Digital twin modelling of ship power and propulsion systems: Application of the open simulation platform (osp). In *2020 IEEE 29th International Symposium on Industrial Electronics (ISIE)*, pages 1265–1270. IEEE, 2020.

[21] Feng Qin, Zhenghe Yan, Peng Yang, Shenglai Tang, and Hu Huang. Corrigendum: Deep-learning-based surrogate model for fast and accurate simulation in pipeline transport. *Frontiers in Energy Research*, 10:1109184, 2022.

[22] Aya Saad and Anne Håkansson. Ramarl: Robustness analysis with multi-agent reinforcement learning-robust reasoning in autonomous cyber-physical systems. *Procedia Computer Science*, 207:3662–3671, 2022.

[23] Aya Saad, Oscar Nissen, Espen Eilertsen, Finn Olav Bjørnson, Tore Norheim Hagtun, Odd-Gunnar Aspaas, Alexia Artemis Baikas, and Sveinung Johan Ohrem. Towards improved visualization and optimization of aquaculture production process. *Procedia Computer Science*, 207:3439–3448, 2022.

[24] Orestis Stavrakidis-Zachou, Nikos Papandroulakis, and Konstadia Lika. A deb model for european sea bass (dicentrarchus labrax): Parameterisation and application in aquaculture. *Journal of Sea Research*, 143:262–271, 2019.

[25] Biao Su, Karl-Johan Reite, Martin Føre, Karl Gunnar Aarsæther, Morten Omholt Alver, Per Christian Endresen, David Kristiansen, Joakim Haugen, Walter Caharija, and Andrei Tsarau. A multipurpose framework for modelling and simulation of marine aquaculture systems. In *International conference on offshore mechanics and arctic engineering*, volume 58837, page V006T05A002. American Society of Mechanical Engineers, 2019.

[26] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? In *Chinese Computational Linguistics: 18th China National Conference, CCL 2019, Kunming, China, October 18–20, 2019, Proceedings*, pages 194–206, 2019.

[27] Hans van de Vis, Jelena Kolarevic, Lars H Stien, Tore S Kristiansen, Marien Gerritzen, Karin van de Braak, Wout Abbink, Bjørn-Steinar Sæther, and Chris Noble. Welfare of farmed fish in different production systems and operations. *The welfare of fish*, pages 323–361, 2020.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[29] BA Williams and Selen Cremaschi. Surrogate model selection for design space approximation and surrogatebased optimization. In *Computer Aided Chemical Engineering*, volume 47, pages 353–358. Elsevier, 2019.

[30] Bianca Williams and Selen Cremaschi. Novel tool for selecting surrogate modeling techniques for surface approximation. In *Computer Aided Chemical Engineering*, volume 50, pages 451–456. Elsevier, 2021.