MDPI

Article

Ultrasonic Sensor Modeling with Support Vector Regression

Duy Ngoc Dang[1], Tri Minh Do[1], Rui Alexandre de Matos Araújo[2], Khang Hoang Vinh Nguyen[3,*], and Can Duy Le[3]

[1] Electrical and Computer Engineering, Vietnamese-German University, Binh Duong, Vietnam; 14445@student.vgu.edu.vn; 15752@student.vgu.edu.vn
[2] University of Coimbra, Institude of Systems and Robotics (ISR-UC), Department of Electrical and Computer Engineering (DEEC-UC), Pólo II, 3030-290 Coimbra, Portugal; rui@isr.uc.pt
[3] Mechatronics and Sensor Systems Technology, Vietnamese-German University, Binh Duong, Vietnam; khang.nhv@vgu.edu.vn; can@vgu.edu.vn
* Correspondence: khang.nhv@vgu.edu.vn; Tel.: +84-77-698-2248

Abstract: This study proposes a novel approach for predicting the output behaviors of the Pepperl+Fuchs 3RG6232-3JS00-PF ultrasonic sensor. The sensor, integrated into the Festo MPS-PA Didactic System, serves to monitor the water level in a tank, facilitating water extraction to bottles delivered via a conveyor belt. This modeling approach represents the initial phase in the creation of a digital twin of the physical sensor, providing the capability for users to observe the sensor's response and forecast its life cycle for maintenance objectives. This study utilizes the Festo MPS-PA Compact Didactic system and support vector regression (SVR) for data acquisition (DAQ), preprocessing, model training with hyper-parameter optimization. The objective of this modeling approach is to establish a digital framework for transition towards Industry 4.0. It holds the potential for creating a digital counterpart of the entire MPS-PA System when combines the proposed sensor modeling technique with computer-assisted design (CAD) software such as Siemens NX in the future. This would enable users to oversee the entire process in a three-dimensional visualization engine such as Tecnomatix Plant Simulation. This research significantly contributes to the comprehension and application of digital twins in the realm of mechatronics and sensor systems technology. It also underscores the importance of digital twins in enhancing the efficiency and predictability of sensor systems. The method used in this paper involves predicting the rate of change (RoC) of the water level and then integrating this rate to estimate the actual water level, providing a robust approach for sensor data modeling and digital twin creation. The result shows a promising 6.99% error percentage.

Keywords: Support Vector Regression; Digital Twin; Virtual Sensor; Kernel Selection; Hyperparameter Optimization.

Citation: D. N. Dang; T. M. Do; R. Araújo; K. H. V. Nguyen; C. D. Le. Ultrasonic Sensor Modeling with Support Vector Regression. *Sensors* **2024**, *1*, 0. https://doi.org/

Received:
Revised:
Accepted:
Published:

1. Introduction

Industrial automation thrives on precise and reliable measurements. Ultrasonic sensors have emerged as a cornerstone technology for diverse tasks, from meticulously gauging liquid levels in tanks to measuring material thickness and determining object proximity with high accuracy. Their operation revolves around emitting high-frequency sound waves and measuring the time it takes for echoes to return. This allows for precise distance calculations and characterization of material properties. Some notable achievements include the development of advanced ultrasonic imaging techniques and the integration of ultrasonic sensors with automated control systems to enhance process efficiency.

Beyond these sensor advancements, machine learning is playing a crucial role in enhancing the performance of industrial measurements using ultrasonic sensors. By analyzing the complex data generated by these sensors, machine learning can identify subtle patterns and trends that would be difficult to discern with traditional methods [1]. Techniques such as neural networks, support vector machine (SVM), and decision trees have been employed

to optimize measurement accuracy and detect anomalies. Furthermore, machine learning models can be trained to predict equipment wear and tear based on sensor data, allowing for scheduled maintenance and preventing unplanned downtime.

This research delves into the exciting realm of data-driven approaches, specifically by leveraging machine learning within a digital twin framework. Digital twins are virtual representations of physical assets and systems that integrate real-time data from sensors and other sources. Traditionally, these systems relied on physical-based models to describe the behaviors of sensors and actuators. These models rely on engineering principles and require significant upfront knowledge of the system dynamics. However, a recent shift is happening in the state-of-the-art of digital twins. There's growing interest in data-driven approaches that leverage real-world data to describe sensor and actuator behavior [2]. This study proposes exploring data-driven approaches within a digital twin framework, aiming to investigate on improving the overall performance and reliability of digital twins across various industrial applications.

The digital twin was constructed within the Siemens TIA Portal software using the Festo MPS PA Didactic System. TIA Portal's WinCC Runtime Advanced serves as the development environment for the PLC program controlling the station, including DAQ from the dosing tank's ultrasonic sensor. This research aims to develop a SVR model for predicting future liquid levels using this sensor data. Data will be collected through TIA Portal and undergo pre-processing to ensure quality for model training. The SVR model will be trained and evaluated using metrics like Mean-squared error (MSE) and R-squared, with hyperparameter tuning employed to optimize prediction accuracy. Importantly, this research explores the potential for real-time control by integrating the trained SVR model within the digital twin. This could allow for using predicted water levels to adjust process parameters in real-time, such as regulating pump operation or valve positions, ultimately optimizing dosing tank liquid levels.

Previous research has explored the application of machine learning techniques in various industrial settings, including predictive maintenance, quality control, and process optimization. For example, developing digital twins to present the framework and workflow of the data-driven models for wind turbines [3], using digital twin technology for production optimization in petrochemical industry [4]. However, the integration of machine learning with digital twins to enhance sensor-based predictions and process control within industrial automation training systems remains a relatively unexplored area. This study aims to contribute to this emerging field by demonstrating the potential of SVR models in predicting liquid levels and exploring the creation of a digital twin framework for the Festo MPS PA bottling station.

The study achieved promising initial results. The SVR model integrated within the Festo MPS PA Didactic System and TIA Portal showed potential for effectively predicting liquid level. Additionally, exploring a data-driven, sensor-agnostic prediction approach and a virtual entity framework within the TIA Portal suggests promising future developments in digital twin technology for industrial automation training systems.

2. Related Work

Digital twins have emerged as a critical technology for enhancing industrial processes through cyber–physical integration. A comprehensive review by Tao et al. [5] provides a comparative analysis of cyber-physical systems (CPS) and digital twins, highlighting the subtle yet important differences between these two concepts. While both technologies aim to achieve seamless interaction between the physical and digital worlds, the study emphasizes the distinct capabilities and applications of digital twins in smart manufacturing. The current research aligns with this analysis by utilizing a digital twin framework, specifically tailored for predictive modeling in process automation, thus extending the scope of digital twin applications beyond traditional CPS.

Kammerer et al. [6] applied digital twins for anomaly detection in manufacturing systems, focusing on real-time data analysis to enhance predictive maintenance strategies.

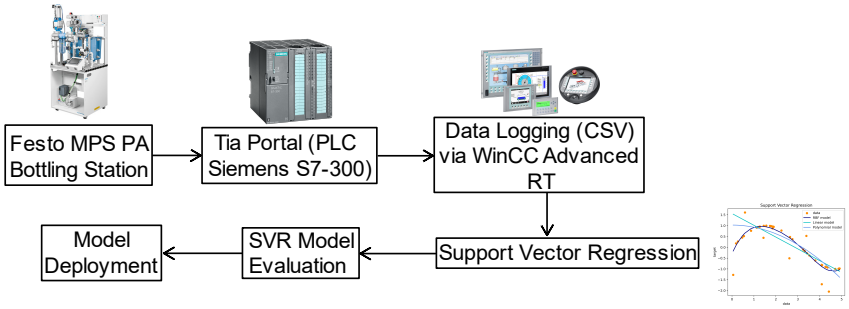


Figure 1. Overview of the proposed modeling method.

Although this approach effectively identified performance deviations, it did not incorporate machine learning models for continuous variable prediction, such as water levels. By integrating SVR to predict the RoC of water levels, the current study addresses this gap, particularly in mitigating error accumulation over time.

Chryssolouris et al. [7] explored digital twin technology for estimating the remaining useful life (RUL) of manufacturing equipment, leveraging physics-based simulation models. This method provided accurate predictions without disrupting operations, showcasing the potential of digital twins in predictive maintenance. However, the approach remained reliant on traditional simulations, whereas the current study adopts a data-driven methodology using radial basis function (RBF) SVR, which enhances prediction accuracy through parameter optimization.

Similarly, Putawa et al. [8] investigated the use of digital twins for visualizing and controlling energy efficiency in manufacturing environments. While effective in system control, the study focused primarily on visualization rather than predictive analytics. The current research shifts towards predictive capabilities, employing SVR within a digital twin framework to forecast water levels accurately, thereby enhancing process control and monitoring.

In summary, this research advances the application of digital twins by integrating RBF SVR for RoC prediction and optimizing SVR parameters to minimize error accumulation, thereby addressing specific challenges in process automation and control. This approach not only builds upon existing studies but also contributes to the broader field of digital twin technology in industrial settings.

3. Methodology

This research employs the Festo MPS-PA Didactic Complete System's bottling station and SVR to establish a machine learning framework. The primary objective of this framework is to predict the water level in the tank, a measurement typically acquired via an installed ultrasonic sensor. An overview of the system setup is illustrated in Figure 1, providing a detailed visual representation of the configuration.

3.1. Festo MPS PA bottling station

The Festo MPS PA system comprises four stations: Filtration, Mixing, Reactor, and Bottling. These stations simulate the water filtration process, providing students with practical knowledge in Process automation and Control theory. This is achieved by controlling multiple variables, including pressure, flow rate, temperature, and water level. This study focuses on the Bottling station, which houses the ultrasonic sensor. The aim is to develop a digital twin of this sensor. Further details of the experimental setup will be discussed in Section 4.

3.2. Human-Machine Interface

The human-machine interface (HMI), designed using Siemens' WinCC Runtime Advanced, enables operators to control processes and monitor real-time data. It also collects

operational data, including timestamps, sensor readings, and actuator values. While a DAQ device can directly obtain these values, the standard in process automation is to use the HMI. This study adheres to these industrial standards, ensuring its applicability in real-world scenarios.

3.3. Support Vector Regression

In this study, SVR is implemented as the machine learning technique to forecast the output of an ultrasonic sensor. The accurate prediction serves as an early warning system for potential failures, which are indicated by discrepancies between actual and predicted values [9]. SVR, a robust supervised machine learning algorithm, is specifically designed for regression tasks [10]. It generates predictions based on a set of input data points. Unlike conventional regression methods that strive to minimize the total error between predicted and actual values, SVR constructs a hyperplane in a high-dimensional space that maximizes the error margin around the most influential data points, referred to as support vectors [10]. This focus on margin minimization reduces the influence of outliers, rendering SVR particularly suitable for applications involving noisy sensor data, such as ultrasonic sensor measurements in industrial settings.

For this study, a RBF kernel was chosen for the SVR model due to its effectiveness in handling non-linear relationships that might exist between sensor readings and water level. The model development utilized the scikit-learn library [11]. The pre-processed sensor data obtained from the HMI was divided into training and testing sets. The training set was used to train the model, while the testing set evaluated the model's ability to generalize to unseen data. Hyperparameter tuning, employing a grid search technique, was conducted to identify the optimal combination of parameters that minimize the MSE on the validation set, by using 3 nested for loops to run through all the possible combinations of hyper-parameter for the best result. The performance of the trained model was subsequently evaluated using metrics like MSE and R-squared on the testing set [12].

3.3.1. Explanation of hyperparameters:

- Cost (C): Controls the trade-off between maximizing the margin and minimizing the training error. A higher C value allows less margin violation, leading to a more complex model.
- Gamma (γ): Influences the width of the RBF kernel function. A smaller gamma results in a wider kernel, leading to smoother decision boundaries.
- Epsilon (ε): Defines the width of the insensitive zone within the epsilon-SVR formulation. Points within this zone are not penalized in the loss function.

By carefully tuning these hyperparameters, the optimal SVR model for predicting water level was determined.

4. Experiment Setup and Implementation

4.1. Festo MPS PA Bottling Station

The Festo MPS PA Bottling Station serves as a valuable training tool in educational settings, specifically designed to introduce students to the intricacies of industrial bottling processes. While not a full-fledged industrial system itself, it effectively simulates the steps involved in filling bottles with liquid. It features the following essential components that mimic a real bottling line (Figure 2):

1. Reservoir tank (Area 19 cm $\times$ 19 cm, Height 34 cm): This tank functions as the primary storage container for the liquid to be bottled. It has a larger capacity compared to the dosing tank and is refilled periodically to maintain a stable liquid supply.
2. Dosing tank (Top diameter: 18 cm, Mid diameter: 13 cm, Bottom diameter: 4 cm, Height 33 cm): This tank acts as a container between the reservoir tank and the bottling line. It has a smaller capacity than the reservoir tank and integrates with an ultrasonic sensor to precisely measure the liquid level in the tank.

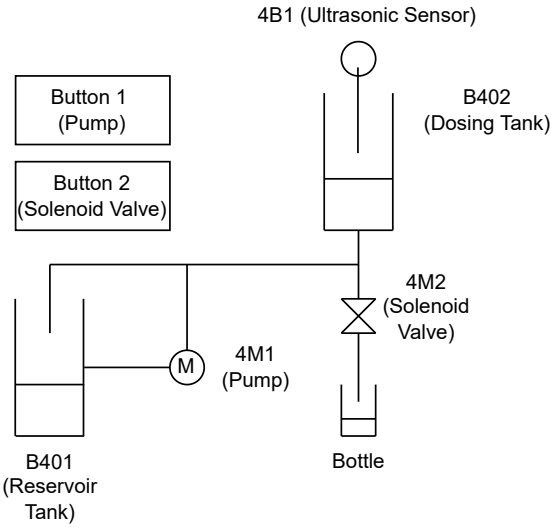


Figure 2. Festo MPS PA Bottling Station.

3. Pepperl+Fuchs Ultrasonic Sensor 3RG6232-3JS00-PF: This sensor leverages ultrasonic technology to achieve continuous, non-contact measurement of the liquid level within the dosing tank. The acquired data plays a critical role in the bottling process by enabling real-time monitoring and control, directly contributing to consistent product volume in the filled bottles by preventing overflows or underfills.
4. Johnson CM30P7-1 Pump: The Festo MPS PA bottling station employs the Johnson CM30P7-1, a compact and efficient centrifugal pump, to ensure a smooth flow of liquids within the system. This pump leverages a rotating impeller to generate centrifugal force, effectively propelling the liquid through the pump housing.
5. Gemü 524D114124DCU solenoid valve: This solenoid valve is utilized for controlled liquid dispensing during the bottling process, and facilitates precise filling through a potential adjustable flow rate mechanism and material compatibility with the processed liquid. Anti-drip design is incorporated to minimize drips or spills after dispensing, promoting cleanliness and reducing product waste.
6. Siemens S7-300 CPU 314C-2PN/DP: The Siemens S7-300 programmable logic controller (PLC) plays a central role in automating the Festo MPS PA Bottling station. It receives signals (buttons) and controls actuators (pump, valve) based on its program. The PLC makes the bottling process automated and flexible.

4.2. Test Cases and Experiment Settings

This section details the design and evaluation procedures employed to assess the SVR model's performance for water prediction in the Festo MPS PA bottling station simulation (Figure 3).

4.2.1. Scenarios.

The test cases mimicked real-world bottling conditions by simulating various water level fluctuations that might occur during the filling process. These scenarios involved in manipulating the two buttons (pump and solenoid valve) and the DAQ button to represent diverse operational conditions.

- Filling cycle variations: Test cases included different buttons press combinations and duration for the pump and valve buttons. This simulated variations in water transfer volumes between tanks and diverse fillings operations.
- DAQ: Users could define the time interval between data points logged in the CSV file via the DAQ button. This allowed for customization based on the desired data granularity for model training.

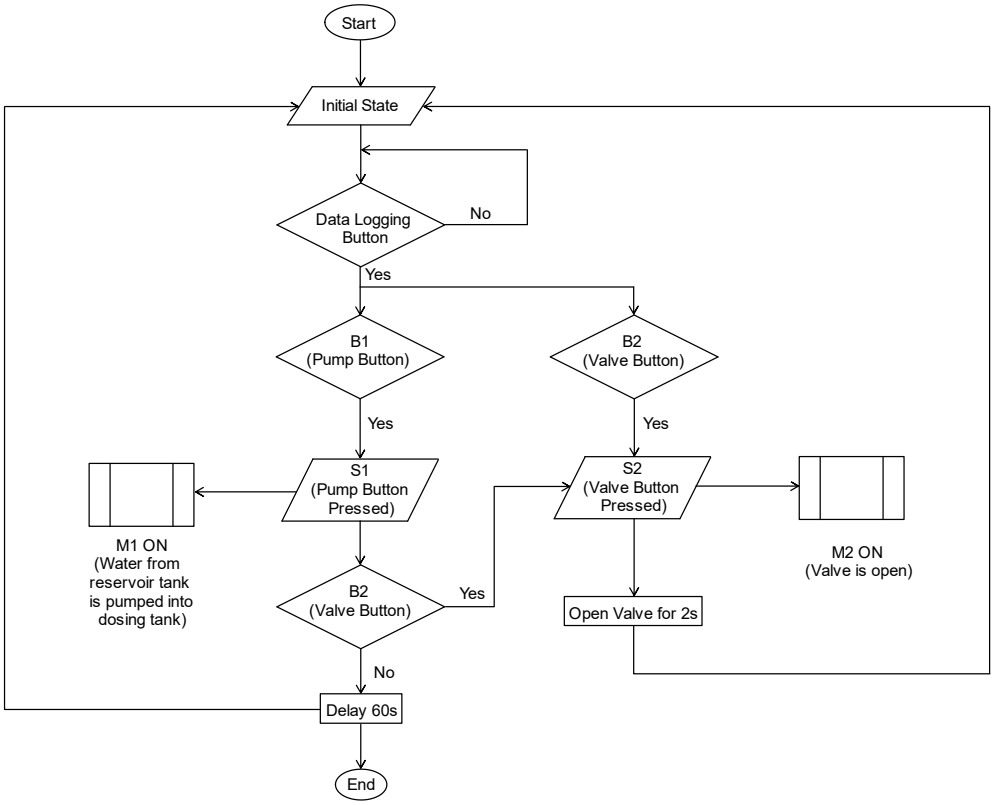


Figure 3. Experiment Flowchart.

4.2.2. Settings

The experiment setup involved a two-pronged approaches:

- **TIA Portal Program**: The operation logic and communication between components were programmed within TIA Portal. This program included functions for:
  - Button control: Responding to user interaction with the pump and valve buttons, triggering virtual button presses within the WinCC environment.
  - Sensor communication: Establishing communication with the ultrasonic sensor to retrieve real-time water level readings.
  - Data transfer: Potentially transferring the collected water level data to WinCC for further processing or visualization.
- **WinCC Advanced Runtime Environment**: WinCC Advanced RT provided the platform for controlling the experiment, acquiring data, and triggering data logging scripts. The functionalities within WinCC were achieved through the Visual basic (VB) scripts.

4.2.3. SCADA design and Data Logging using VB Script.

The experiment leveraged a supervisory control and data acquisition (SCADA) design approach within WinCC Advanced Runtime to control the simulation and acquire data for model development. VB Scripts played a vital role in this SCADA design, acting as the interface between the HMI and the physical components.

- **SCADA design elements**:
  - HMI control: The VB Scripts provided functionalities displayed on the HMI, such as buttons for pump and valve activation, and a DAQ button.
  - DAQ: The scripts interacted with the ultrasonic sensor through WinCC tags, retrieving water level readings at a defined interval.

  - Data logging: Upon pressing the DAQ button, the script triggered data logging into a CSV file.
- **VB scripts functionalities**:
  - Simulation and Control: The scripts facilitated precise control over the simulated filling cycles by automatically triggering virtual button presses based on the defined scenarios.
  - DAQ: The scripts interacted with the ultrasonic sensor through WinCC tags to retrieve real-time water levels at specified intervals throughout the simulated filling cycles.
  - Data logging: When pressed, the DAQ button will activate the scripts to retrieve the collected data and write it along with timestamps to a user-defined location in a CSV format. This enabled easy import into data analysis tools for model training and evaluation.

4.3. Format of Data

The VB scripts generated a CSV file for model training and evaluation. This file offers a well-defined format for efficient processing by the SVR model. Each column of data is separated by a semicolon (;) delimiter. The data format includes:

- Timestamps: Captures the time of each data point (e.g., YYYY-MM-DD HH:MM:SS) for time-based analysis.
- Water Level Reading: Represents the real-time water level measurement (milliliters) - the target variable for prediction.
- Input Features of the VB screen in WinCC:
  - **Pump**: Toggle the ON and OFF state of the 4M1 Pump;
  - **Valve**: Toggle the ON and OFF state of the 4M2 Valve;
  - **Start data logging**: Start the VB script that logs the data on the WinCC server side;
  - Other input data is controlled by directly modifying the values in TIA Portal's HMI tag table.

By adhering to this structured format with semicolon delimiters, the CSV file provides a well-organized dataset suitable for machine learning model training and evaluation.

5. Model Development

5.1. Lagrange Duality

Lagrange duality is a fundamental concept in convex optimization and plays an important role in SVR algorithm [13]. An optimization often has a standard form as follows:

$$\begin{aligned} &\text{minimize } f_0(x) \\ &\text{subject to } \begin{cases} f_i(x) \leq 0, i = 0, ...m \\ h_i(x) = 0, i = 0, ...p \end{cases} \end{aligned} \tag{1}$$

A Lagrangian approach is to optimize equation (2) instead:

$$\inf_x \mathcal{P}(x) = \inf_x \sup_{\lambda \geq 0} \mathcal{L}(x, \boldsymbol{\lambda}) = \inf_x \sup_{\lambda \geq 0} \left(f(x) + \sum_k \lambda_k f_i(x) \right) \tag{2}$$

where $\mathcal{L}(x, \boldsymbol{\lambda}) = f(x) + \sum_k \lambda_k f_i(x)$, $\mathcal{P}(x) = \sup_{\lambda \geq 0} \mathcal{L}(x, \boldsymbol{\lambda})$.

Equation (2) is referred to as the Primal Lagrangian equation. The second alternative to the Primal function is the Dual Form equation:

$$\sup_{\lambda \geq 0} \mathcal{D}(\boldsymbol{\lambda}) = \sup_{\lambda \geq 0} \inf_x \mathcal{L}(x, \boldsymbol{\lambda}) \leq \inf_x \sup_{\lambda \geq 0} \mathcal{L}(x, \boldsymbol{\lambda}) = \inf_x \mathcal{P}(x) \tag{3}$$

where $\mathcal{D}(\boldsymbol{\lambda}) = \inf_x \mathcal{L}(x, \boldsymbol{\lambda})$.

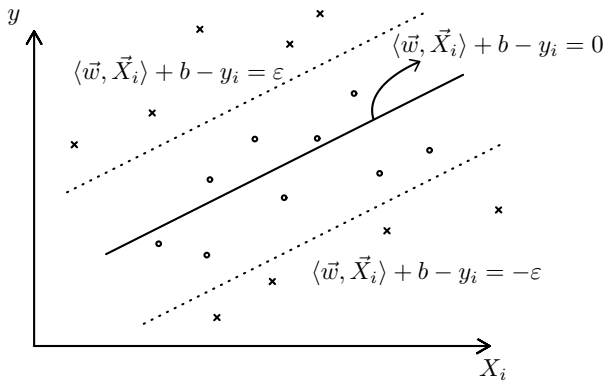


Figure 4. Regression of a linear kernel.

5.2. Support Vector Machine with Linear Kernel

Given a linear kernel, a regression line will have a general linear function in the form of equation (4).

$$\langle \boldsymbol{w}, \boldsymbol{X}_i \rangle + b - y_i = 0 \tag{4}$$

The objective of SVR is to determine the vector $\boldsymbol{w}$ such that the resulting linear function ensures most of the data points $(\boldsymbol{X}_i, y)$ remain within a specified ε tolerance. Figure 4 shows the general presentation of the kernel in the case that $\boldsymbol{w}$ and $\boldsymbol{X}_i$ only have 1 dimension, but in application, both of these vectors can have a much higher dimensions that cannot be represented in a Cartesian coordinate.

The aim of SVR now becomes equation (5)

$$\begin{aligned} &\text{minimize } \frac{1}{2}\|\boldsymbol{w}\|^2 \\ &\text{subject to } \begin{cases} y_i - \langle \boldsymbol{w}, \boldsymbol{X}_i \rangle - b \leq \varepsilon \\ \langle \boldsymbol{w}, \boldsymbol{X}_i \rangle + b - y_i \leq \varepsilon \end{cases} \end{aligned} \tag{5}$$

Equation (5) assumes that the convex optimization for a linear kernel function $f(\boldsymbol{X}) = \langle \boldsymbol{w}, \boldsymbol{X} \rangle + b$ with an error of ε is feasible. Sometimes, this may not be the case and some amount of errors needs to be taken into account. Hence the slack variables ξ_i and ξ_i^* need to be used [10].

$$\begin{aligned} &\text{minimize } \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ &\text{subject to } \begin{cases} y_i - \langle \boldsymbol{w}, \boldsymbol{X}_i \rangle - b \leq \varepsilon + \xi_i \\ \langle \boldsymbol{w}, \boldsymbol{X}_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \tag{6}$$

The parameter C determines the trade-off between the flatness of f and determines how much amount of deviation from the target error ε is tolerated [10].

5.3. Gaussian kernel

RBF kernel regression, or sometimes called Gaussian kernel regression, is a popular algorithm that is widely used in SVM tasks, or in this case, SVR [14].

The kernel function for 2 set of parameters is defined in equation (7)

$$K(\boldsymbol{X}, \boldsymbol{X}_i) = \exp\left(-\gamma \|\boldsymbol{X} - \boldsymbol{X}_i\|^2\right) \tag{7}$$

where $\gamma = \frac{1}{2\sigma^2}$, σ is the standard deviation parameter of the Gaussian curve, which also determines the width of the Gaussian kernel [14].

The weight w_i used in support vector regression is determined based on the similarity between the kernel K of data i and the rest of the data points.

$$w_i = \frac{K(\boldsymbol{X}, \boldsymbol{X}_i)}{\sum_{j=1}^{N} K(\boldsymbol{X}, \boldsymbol{X}_j)}. \tag{8}$$

Implementing the kernel into an SVM system, equation (9) is obtained

$$\begin{aligned} &\text{minimize } \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*), \\ &\text{subject to } \begin{cases} y_i - \sum_{j=1}^{N} \alpha_j K(\boldsymbol{X}_i, \boldsymbol{X}_j) \leq \varepsilon + \xi_i, \\ \sum_{j=1}^{N} \alpha_j K(\boldsymbol{X}_i, \boldsymbol{X}_j) - y_i \leq \varepsilon + \xi_i^*, \\ \xi_i, \xi_i^* \geq 0. \end{cases} \end{aligned} \tag{9}$$

5.4. Hyper-parameters explanation

The C parameter is a scalar that controls the penalty for the values that deviate from the predicted regression curve. The higher the value C is, the more noticeable the error is [11]. In contrary, a smaller value of C allows for a larger margin and more tolerance for errors, which can lead to a simpler model with potentially higher bias but lower variance.

The γ value controls the width of the Gaussian kernel function used for the regression process. This parameter defines the extent to which a single training example influences the model's decision boundary [11]. A lower γ results in a wider influence, making each training example affect a larger area of feature spaces.

ε is the margin of tolerance around the predicted curve. If a data point is positioned within this margin, the point will have 0 error penalty [11]. This margin will act as an error threshold. Similar to the C parameter, a larger ε will result in a higher bias and lower variance predicted curve.

5.5. Model explanation

The input layer of the network consists of six variables:

Variable	Data type	Range
Time	Integer, in seconds	0 to $+\infty$
Button 1	Boolean	ON and OFF
Button 2	Boolean	ON and OFF
Valve State	Boolean	ON and OFF
Pump State	Boolean	ON and OFF
Pump Analog	Integer	0 to 27648

Upon initial observation using Microsoft Excel's graphing tool, utilizing the RoC of the water level as training data rather than the water level itself appears to be more effective for the labeling process. This is due to the direct correlation observed between the valve state (0 and 1), Button and Pump values, and the RoC of the water level.

The discrete RoC of the parameter is calculated as follows:

$$\frac{\Delta y}{\Delta t} = \frac{y_i - y_{i-1}}{1\,\text{s}} = y_i - y_{i-1}. \tag{10}$$

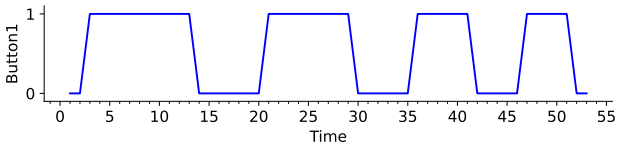
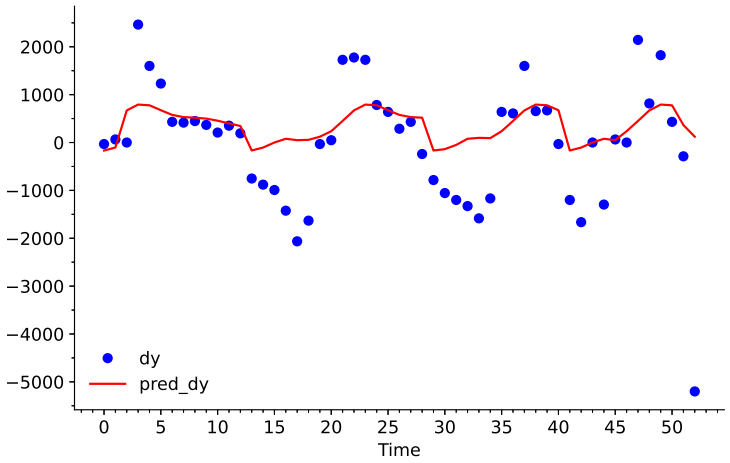


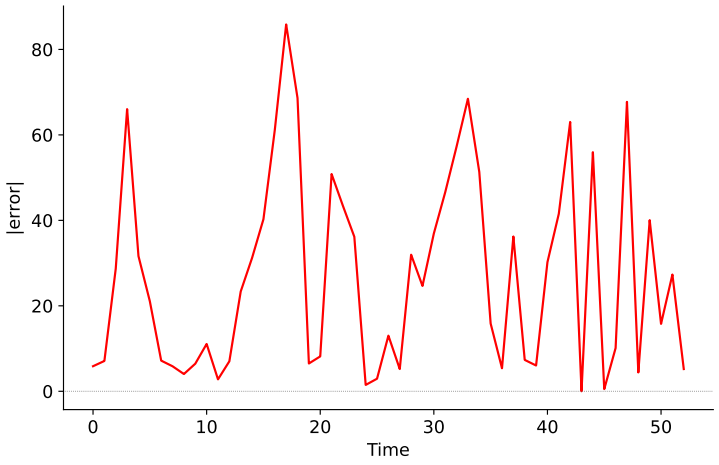
Figure 5. Value of `Button1` with respect to `Time` in seconds.

The `Time` variable is relabeled manually with respect to `Button1` (Fig. 5) to make the learning process more effective. This is due to Button1 having the most influence over the predicted data. Each time `Button1` undergoes from 0 to 1 and to 0 again, the `Time` variable is reset to 0, creating an ON and OFF cycle, which is equal to splitting 1 dataset into many smaller datasets for better pattern recognition.

6. Model Evaluation and Chosen hyperparameters



(a)



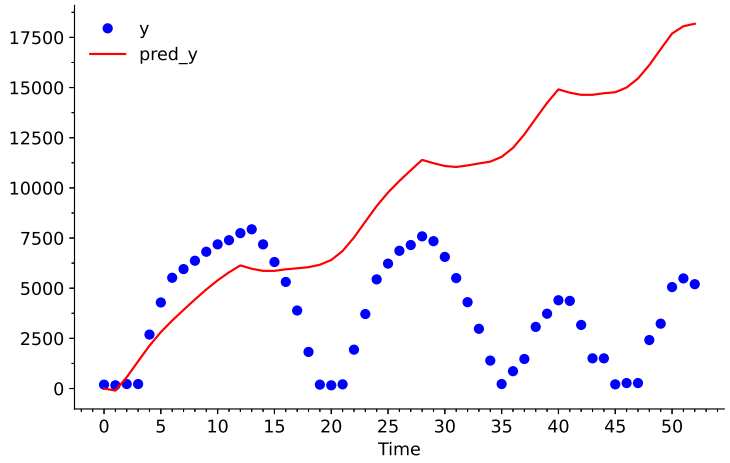
(b)

Figure 6. (a) Plot of `dY` and the predicted curve `pred_dY` using conventional hyperparameters and (b) the Error in percentage with respect to `Time` in seconds.

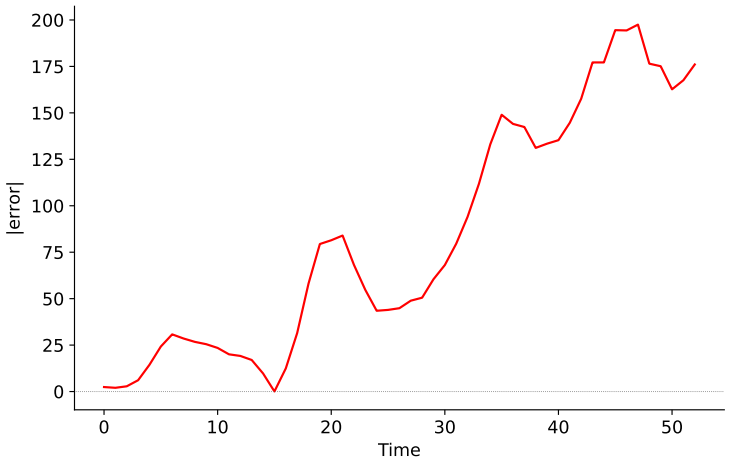
In this experiment, the true water level measured using the ultrasonic sensor will be assigned to `Y`, and the water level predicted by the algorithm will be stored to `pred_Y`.

Similarly, the RoC of the water level derived from `Y` is stored in `dY` and the predicted RoC of the water level (which will be used to calculate `pred_Y`) is assigned to `pred_dY`. The experiment was performed 5 times and produce no significant different (less than 0.01%) for each set of hyperparameter C, ε, and γ.

When the predicted curve undergoes the integration process later on in the network, any slight deviation between `dY` and `pred_dY` will stack up overtime. Evidently, even though the error is not observed to be a problem in Fig. 6, the difference between `Y` and `pred_dY` keeps increasing continuously in Fig. 7.



(a)



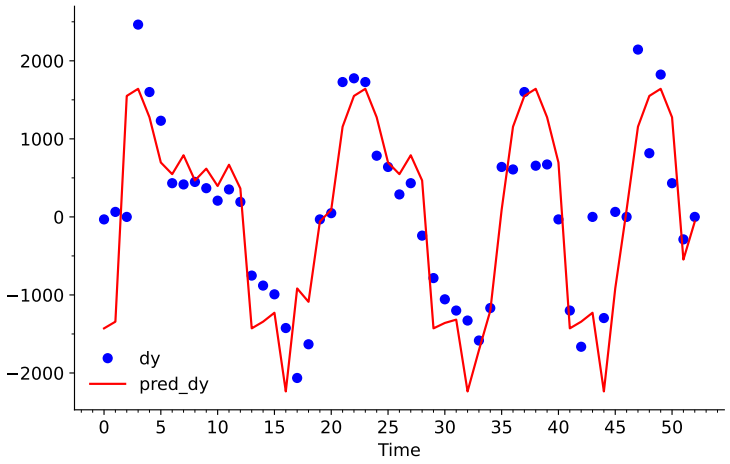
(b)

Figure 7. (a) Plot of `Y` and the predicted curve `pred_Y` using conventional hyperparameters and (b) the Error in percentage with respect to `Time` in seconds.

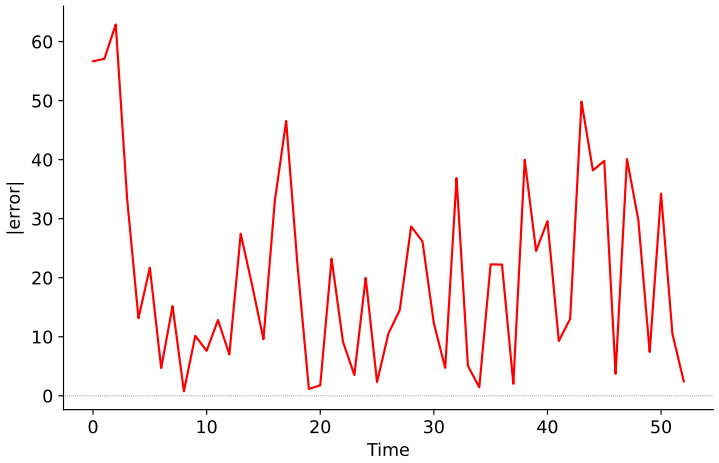
This makes it more effective to use a higher C and lower ε value for the training process. Lowering ε will decrease the tolerance margin for the error value, and increasing the C will increase the penalty of each predicted value that deviates further from the ε-margin. This will force `pred_dY` to be much closer to the value of `dY`. In turn, this will also increase the variance of `pred_dY` as discussed in Section 5.4. This should not be a problem in this particular use case because the integration process will smooth-out most of the variance as long as dY do not change rapidly from positive to negative values.

The chosen hyperparameters for the problem is: $C = 320$, $\gamma = 0.72$, $\varepsilon = 1.3$.

Upon comparison between Fig. 6 and Fig. 8, it becomes clear that the adjusted C, γ, and ε values have greatly decreased the bias between the dataset and the predicted curve. Particularly, values from $t = 2 \rightarrow 5, 12 \rightarrow 25, 27 \rightarrow 31$ s have a larger influence over `pred_dY`. This helps largely reduce the deviation in the integration process as shown in Fig. 9.



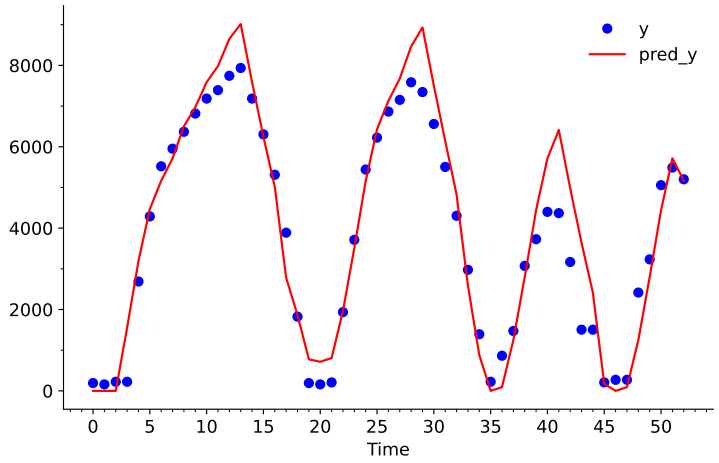
(a)



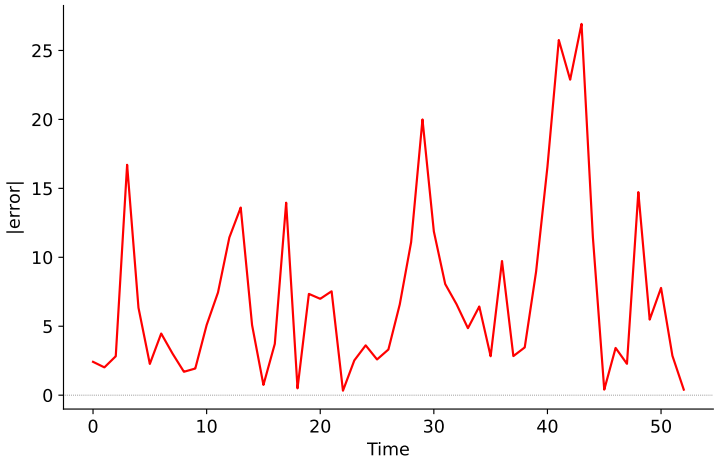
(b)

Figure 8. (a) Plot of `dY` and the predicted curve `pred_dY` using adjusted hyperparameters and (b) the Error in percentage with respect to `Time` in seconds.

`pred_dY` may have a high variation, but a large or small positive value still results in an increase in `dY`. Similarly, a large or small value of negative `pred_dY` will result in a decrease in `dY`. This characteristic can be demonstrated by taking the average error of Fig. 8b and Fig. 9b. Even though the SVR model showed an average 18.38% error rate for predicting RoC of the water level, but only an average 6.99% error rate for predicting the actual water level after integration.



(a)



(b)

Figure 9. (a) Plot of `Y` and the predicted curve `pred_Y` using adjusted hyperparameters and (b) the Error in percentage with respect to `Time` in seconds.

7. Conclusion

In conclusion, the evaluation of the sensor output prediction model, the optimization of hyperparameters, and the utilization of derivation and integration have provided increase in accuracy for the sensor behavior predictions.

Subsequent to this experiment, it is advised to integrate more parameters into the system. Specifically, the application of simulation coupling could optimize computational resources and reduce inference time [15]. This addition would introduce another layer to the network and alleviate the drifting error that occasionally cannot be eradicated solely by predicting the derivative [16]. Furthermore, the incorporation of sensor fusion algorithms, such as the Kalman filter, could significantly enhance the model's precision by consolidating data from diverse sources [17]. This strategy offers a more resilient and precise prediction model for future applications.

Lastly, to visualize and analyze the system holistically, a digital twin representation can be constructed using CAD software like Siemens' NX for the ultrasonic sensor and a three-dimensional visualization engine like Tecnomatix Plant Simulation for the entire Festo MPS PA system. This approach could serve as a foundation for digitizing various manufacturing processes [18].

7.0.1. Acknowledgements

This research is funded by Vietnamese-German University (Vietnam) under grant number DTCS-2023-002.

Author Contributions: Conceptualization, C. D. Le; methodology, T. M. Do and D. N. Dang; software, T. M. Do, D. N. Dang, and K. H. V. Nguyen; validation, T. M. Do; formal analysis, T. M. Do; investigation, T. M. Do and D. N. Dang; resources, K. H. V. Nguyen; data curation, T. M. Do and D. N. Dang; writing—original draft preparation, T. M. Do and D. N. Dang; writing—review and editing, R. Araújo, C. D. Le, and K. H. V. Nguyen; visualization, T. M. Do; supervision, C. D. Le; project administration, K. H. V. Nguyen; funding acquisition, K. H. V. Nguyen. All authors have read and agreed to the published version of the manuscript.

Acknowledgments: This research is funded by the Vietnamese-German University Research Funding under grant number DTCS-2023-002.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SVR	Support Vector Regression
SVM	Support Vector Machine
RBF	Radial Basis Function
DT	Digital Twin
RUL	Remaining Useful Life
CPS	Cyber-Physical Systems
RoC	Rate of Change
HMI	Human-Machine Interface
DAQ	Data Acquisition
MSE	Mean-Squared Error
PLC	Programmable Logic Controller
VB	Visual Basic
SCADA	Supervisory Control and Data Acquisition
CAD	Computer-Assisted Design

References

1. Kaneko, H.; Funatsu, K. Application of online support vector regression for soft sensors. *AIChE Journal* **2013**, *60*, 600–612. https://doi.org/10.1002/aic.14299.
2. Ayadi, R.; El-Aziz, R.M.A.; Taloba, A.I.; Aljuaid, H.; Hamed, N.O.; Khder, M.A. Deep Learning–Based soft sensors for improving the flexibility for automation of industry. *Wireless Communications and Mobile Computing* **2022**, *2022*, 1–10. https://doi.org/10.1155/2022/5450473.
3. Hu, W.; Zhang, T.; Deng, X.; Liu, Z.; Tan, J. Digital twin: a state-of-the-art review of its enabling technologies, applications and challenges. *Journal of Intelligent Manufacturing and Special Equipment* **2021**. https://doi.org/10.1108/JIMSE-12-2020-010.
4. Thelen, A.; Zhang, X.; Fink, O.; Lu, Y.; Ghosh, S.; Youn, B.D.; Todd, M.D.; Mahadevan, S.; Hu, C.; Hu, Z. A comprehensive review of digital twin — part 1: modeling and twinning enabling technologies. *Structural and Multidisciplinary Optimization* **2022**, *65*, 354. https://doi.org/10.1007/s00158-022-03425-4.
5. Tao, F.; Qi, Q.; Wang, L.; Nee, A. Digital Twins and Cyber–Physical Systems toward Smart Manufacturing and Industry 4.0: Correlation and Comparison. *Engineering* **2019**, *5*, 653–661. https://doi.org/https://doi.org/10.1016/j.eng.2019.01.014.
6. Kammerer, K.; Hoppenstedt, B.; Pryss, R.; Stökler, S.; Allgaier, J.; Reichert, M. Anomaly Detections for Manufacturing Systems Based on Sensor Data—Insights into Two Challenging Real-World Production Settings. *Sensors* **2019**, *19*, 5370. https://doi.org/10.3390/s19245370.
7. Aivaliotis, P.; Georgoulias, K.; Chryssolouris, G. The use of Digital Twin for predictive maintenance in manufacturing. *International Journal of Computer Integrated Manufacturing* **2019**, *32*, 1067–1080. https://doi.org/10.1080/0951192x.2019.1686173.
8. Putawa, R.A.; Wardana, A.N.I.; Tenggara, A.P. Metaverse-based water level simulator for the Festo MPS PA workstation. *Journal of Physics Conference Series* **2023**, *2673*, 012008. https://doi.org/10.1088/1742-6596/2673/1/012008.
9. Grinblat, G.L.; Uzal, L.C.; Verdes, P.F.; Granitto, P.M. Nonstationary regression with support vector machines. *Neural Computing and Applications* **2014**, *26*, 641–649. https://doi.org/10.1007/s00521-014-1742-6.
10. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Statistics and Computing* **2004**. https://doi.org/10.1023/B:STCO.0000035301.49549.88.

11. F., P.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.
12. Frost, J. *Regression Analysis: an intuitive guide for using and interpreting linear models*; Statistics, 2020.
13. Boyd, S.P.; Vandenberghe, L. *Convex Optimization*; Vol. 51, Cambridge University Press, 2004; pp. 1859–1859.
14. Bishop, C. *Pattern Recognition and Machine Learning*; Springer New York, 2006. https://doi.org/10.1007/978-0-387-45528-0.
15. Edtmayer, H.; Brandl, D.; Mach, T.; Schlager, E.; Gursch, H.; Lugmair, M.; Hochenauer, C. Modelling virtual sensors for real-time indoor comfort control. *Journal of Building Engineering* **2023**, *67*, 106040. https://doi.org/10.1016/j.jobe.2023.106040.
16. Shacham, M.; Brauner, N. Minimizing the Effects of Collinearity in Polynomial Regression. *Industrial & Engineering Chemistry Research* **1997**, *36*, 4405–4412. https://doi.org/10.1021/ie970236k.
17. Petersen, C.D.; Fraanje, R.; Cazzolato, B.S.; Zander, A.C.; Hansen, C.H. A Kalman filter approach to virtual sensing for active noise control. *Mechanical Systems and Signal Processing* **2008**, *22*, 490–508. https://doi.org/10.1016/j.ymssp.2007.06.007.
18. Lyu, Z.; Fridenfalk, M. Digital twins for building industrial metaverse. *Journal of Advanced Research* **2023**, p. S2090123223003594. https://doi.org/10.1016/j.jare.2023.11.019.