

Development of an Integrated Digital Twin Framework for Online Monitoring and Anomaly Detection

Lauren Bailey¹, Jamie Coble^{1,2} and Xingang Zhao¹

¹University of Tennessee, Knoxville, Tennessee, 37996, U.S.A

lbaile44@vols.utk.edu

jamie@utk.edu

xzhao47@utk.edu

²University of Wisconsin—Madison, Madison, Wisconsin, 53706, U.S.A

jcoble@wisc.edu

ABSTRACT

Digital Twins (DTs) are emerging as an important enabling technology in the nuclear field, as their development leverages the ongoing digitalization of nuclear systems and offers many advantages for optimizing nuclear operations. Nuclear DTs can be used for predictive maintenance, enhanced operations, improved scheduling, cradle-to-grave lifecycle management, decision support, and more. While the advantages are clear, continued DT development is needed, as many gaps remain, including technological and regulatory standardization, validation and verification, and digital-to-physical communication. To help close these gaps, a forced-water-flow experimental system at the University of Tennessee, Knoxville (UTK) was used to develop a complete, integrated DT, offering an end-to-end use case in the nuclear sector and related disciplines. For this work, a Long Short-Term Memory (LSTM) network was used for forecasting, and an interquartile range (IQR) LSTM hybrid method was used for anomaly detection. This was all interfaced with LabVIEW, enabling real-time data analysis and forecasting. A GUI was then created to provide the operator with a clear indication when a system anomaly is detected, enabling better insights during operation. The resulting platform demonstrates an end-to-end proof of concept for integrating data acquisition, machine learning-based forecasting and anomaly detection, and operator interaction within an experimental DT for various nuclear and related engineering systems.

1. INTRODUCTION

Digital Twins (DTs) offer significant potential for optimizing operations across various fields. DTs are a virtual equivalent of a physical system, shaped by two-way communication (Huang et al., 2025). From healthcare services to manufacturing and operations, DTs have significant potential to enable continuous monitoring and system analysis throughout the lifecycle of a system or product (Attaran & Celik, 2023). While DT implementation spans numerous fields, this work focuses on its development and application within the nuclear sector. This focus is motivated by the overarching long-term goals of this work, including the development of nuclear DTs and the advancement of DT framework standardization in the nuclear field. Development of DTs for nuclear reactors, especially next-gen reactors, is becoming a highly sought-after advancement, but exploration of DTs for aspects of the fuel cycle (front- and back-end), fusion technologies, nuclear materials manufacturing, and more is also a major area of growth (Mengyan et al., 2024). DTs offer a way to bridge nuclear technology into modern operations by enhancing and embracing digitalization, optimizing the efficiency of our nuclear reactors, and ensuring the continued safe and efficient operation of nuclear-related manufacturing and experiments (Bandala et al., 2024).

This work integrates the key components required to develop a comprehensive DT framework for an experimental forced-water-flow system, with an emphasis on creating a transferable methodology applicable to a range of nuclear-related systems. The resulting DT enables real-time system monitoring and anomaly detection, providing enhanced insight and situational awareness for operators. This work includes the operator as part of the physical system and utilizes a human-machine interface (HMI) to support the

Lauren Bailey et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

<https://doi.org/10.36001/IJPHM.2026.v17i2.5076>

digital-to-physical control feedback. The paper is structured to first present a broad overview of DTs and associated data-driven methodologies used in the field. This is followed by a detailed description of the methodology used to develop the digital twin for the physical system and concludes with a demonstration of the complete framework and its implementation in the presented use case.

2. BACKGROUND

2.1. Digital Twins

DTs are virtual representations of a physical system that behave in the same way as the physical system. However, what differentiates them from a typical simulation is the two-way communication that enables real-time data flow to the digital system and, in turn, the transfer of insights, suggestions, or commands back to the physical system. This communication enables a comprehensive ecosystem for the facility. A basic outline of a digital twin for a nuclear system is illustrated in Figure 1 (Nuclear Regulatory Commission (NRC), 2025). The digital system may combine data-driven methods, physical models, and various data analysis techniques. The digital model then sends actions and recommendations back to the physical system. This can include diagnostics and prognostics, operations and maintenance suggestions, or control commands.

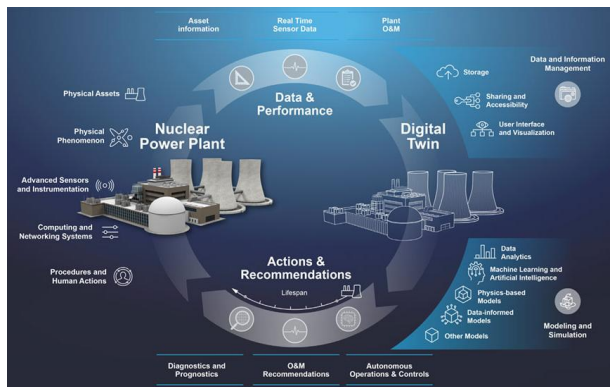


Figure 1. Overview of a nuclear digital twin (Nuclear Regulatory Commission (NRC), 2025)

DTs were first proposed in 2003, and the concept has since evolved into what we know today as a digital twin. (Attaran & Celik, 2023). Since then, many industries have adopted the idea and utilized it to support more efficient and effective system lifecycle management and operations. DT integration has been widely adopted in fields such as aviation, agriculture, and manufacturing. These DTs are built for various purposes, including, but not limited to, anomaly detection, operations and maintenance optimization, quality control, autonomous control systems, and cradle-to-grave design support (Zio & Miqueles, 2024). Since standardized terminology for DTs does not exist, the term is widely used, often without a clear definition; many DT developments merely create a digital shadow, which is a connection from

the physical to the digital system, but lacks the secondary flow of information from the digital system back to the physical system (Crespi et al., 2023). A true DT enables bidirectional information flow, allowing the digital system to influence the physical system and vice versa. Since this work is conducted in support of nuclear engineering development, the focus and definition of the DT used here are also guided by definitions from the U.S. Nuclear Regulatory Commission (NRC) and related organizations, ensuring that the advancements made here are applicable across the broader nuclear industry. In the context of this work, a DT is a digital representation of a physical system that behaves like the real system and enables two-way communication between the two. This definition is consistent with that used by the NRC and by industries with more mature DT infrastructure (AIAA Digital Engineering Integration Committee, 2020; Barricelli et al., 2019; Nuclear Regulatory Commission (NRC), 2025).

DT architectures vary according to the intended application, system scope, available data, and degree of automation. At the modeling level, DTs may use physics-based models, data-driven models, or hybrid approaches that combine mechanistic understanding with operational data. At the architectural level, they may be centralized around a system-level model, distributed across component-level twins, or organized hierarchically to connect component-, subsystem-, and system-level representations. Gregory et al. (2026) emphasize modeling methods and digital thread opportunities to ensure a cohesive, well-maintained relationship among all aspects of a digital twin throughout its life cycle. Although the appropriate architecture varies by application, a comprehensive DT must accommodate heterogeneous models and supporting subsystems, including data management, device communication, human-machine interfaces, and bidirectional communication capabilities (Gregory et al., 2026).

While DTs have become a useful technology in many fields, numerous fundamental research gaps remain as this technology becomes more integrated and advanced. Prior work specifically examined DTs in the advanced nuclear reactor domain and identified major gaps in technological and regulatory standardization, validation and verification, large-scale integration, and two-way communication (Bailey & Zhao, 2025). These gaps were identified in the related nuclear literature, and many also apply to DTs in energy systems more generally. Integrating DTs into standardized use in the energy sector remains a challenge. As mentioned previously, the technological and regulatory gaps are major considerations for this technology and require further standardization if it is to be fully integrated into the nuclear field, as accepted by the NRC or other regulatory bodies. The NRC has published various technical reports that reflect its views and support for DT development, but it is important that these be further developed as the technology expands and that they be followed to ensure that nuclear DTs can be utilized within the regulatory process (Yadav, Eskins, et al.,

2022; Yadav, Wells, et al., 2022). Validation and verification remain significant challenges, particularly for advanced reactors and other nuclear systems still in the design phase. Without a physical system to collect real operational data, these models cannot be fully validated; meaning that while the modeling and virtual framework can progress, true digital twin status and guaranteed fidelity are unattainable until the physical system exists and data becomes available. Two-way communication is also a gap, as much of the current nuclear DT literature is developing digital shadows with no communication from the digital system back to the physical system. This warrants further exploration, as bidirectional communication is a major draw of DTs. There are other gaps indicating a need for continued development of this technology, and research is ongoing across many sectors of the nuclear field (Huang et al., 2025).

While there are gaps in this technology, there have been many recent developments, particularly by Idaho National Laboratory (INL) and its collaborators at Idaho State University (ISU), as well as at Purdue University and the University of Texas (UT) at Austin (Finan et al., 2024; Theos, 2023; Luciano et al., 2025). INL and ISU have developed a DT of the AGN-201 reactor on ISU's campus. The AGN-201 is a 1960s-era system, relatively simple compared to a nuclear plant. This reactor operates at up to 5 W and has no coolant flow. However, this has enabled the development of methodologies for real-time monitoring and anomaly detection, as well as remote monitoring and control capabilities (Stewart et al., 2025). Using advanced INL-developed software tools, such as DeepLynx and Jester, the ISU AGN-201 DT provides a robust framework for nuclear research reactor DTs (Hergesheimer et al., 2026). Additional DT development is underway at Purdue University's PUR-1 research reactor. As a result, they have developed a digital twin to connect with their fully digitized instrumentation and control system (Theos, 2023). A major focus of their DT is cybersecurity, testing, and developing techniques to improve cyberattack prevention and detection (Prantikos et al., 2022). Lastly, UT-Austin has also made strong progress in the nuclear DT space, including the development of a digital shadow for its research reactor, the TRIGA II. They continue to advance toward the development of a full DT and have a focus on web-based DT platforms (Seo et al., 2026). There are many other digital twin or digital shadow works within nuclear, but the gaps discussed above remain, hence the need for a pathway towards further technological development and the efforts towards a transferable framework.

2.2. Data-Driven Techniques

2.2.1. Time-Series Machine Learning

The modeling techniques used to develop a DT's digital system vary depending on the intended application. In this work, the primary objectives are system forecasting and anomaly detection; therefore, the emphasis is on methods that

support these tasks. Specifically, this work employs long short-term memory (LSTM) networks.

Neural networks (NNs) are widely used in machine learning (ML); however, conventional feedforward NNs do not retain information from previous inputs, making them less suitable for sequential data. Sequential data poses a unique challenge for ML techniques because not only do features at each time step correlate, but they also correlate over time. To address this lapse, recurrent neural networks (RNNs) were developed, allowing information to be retained from previous time steps. LSTMs are an advanced RNN architecture and are the primary modeling technique used in this work.

To make accurate predictions while retaining information from past time steps, RNNs maintain a hidden state that stores information from previous time steps, enabling the model to reference it when making its next prediction. RNNs are a strong model for time-series data; however, they can suffer from exploding or vanishing gradients. This occurs because the gradient is repeatedly multiplied during backpropagation through time (Bengio et al., 1994). This disadvantage is mitigated by more advanced time-series ML techniques built on the RNN architecture, which use logic gates to better control memory. Two of these advanced methods are Gated Recurrent Units (GRUs) and the aforementioned LSTMs. GRUs have a reset and an update gate, whereas LSTMs have an input, output, and forget gate (Hochreiter & Schmidhuber, 1997). This improved memory control makes LSTMs and GRUs more widely used in forecasting applications. The basic workings of these three models are shown in Figure 2.

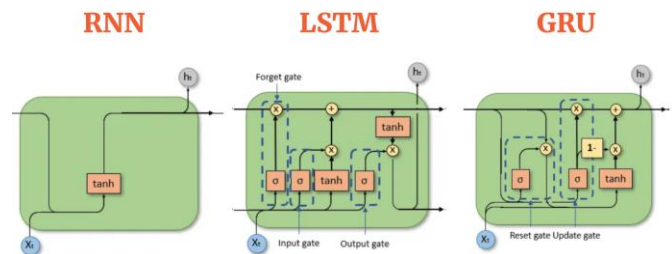


Figure 2. Time series machine learning models (GeeksforGeeks, 2023)

For these time-series ML models, different architectures can be used, depending on the objective. Figure 3 shows a simple illustration of some of the possible architectures. The first option is the most straightforward and is known as a 'one-to-one', in which you use one feature for both training and forecasting. Then comes the 'one-to-many', where you train it on one feature or input and have it predict/forecast multiple features. 'Many-to-one' is the reverse, where you have multiple inputs and predict one output. There are then two types of 'many-to-many' approaches; the first is where the inputs are not all the same as your outputs, meaning you are asking your model to predict a new feature. The second approach is more classical, in which the inputs are forecast; this is the approach used throughout this work.

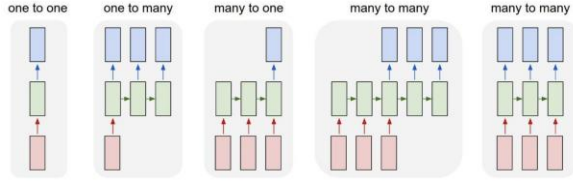


Figure 3. Time series ML input to output architectures (Dhee, 2023)

Several techniques have been adapted for time-series applications, including 1-D convolutional neural networks (CNNs), Transformer models, and others. These models will not be explored here, but related research supports their use.

2.2.2. Anomaly Detection

An anomaly in a time-series signal is a deviation from expected behavior. This may manifest as a point anomaly, a gradual system drift, or other fault types. Time-series anomaly detection is a challenging problem due to temporal dependencies, as past values influence future ones, making it important to optimize detection strategies that consider behavior over a meaningful time window (Wang et al., 2025).

Anomaly detection methods can be broadly categorized as statistical, rule-based, or data-driven machine learning approaches. Several techniques are available for time-series analysis, including simple thresholding, sequential probability ratio testing (SPRT), LSTM autoencoder thresholding, and more advanced hybrid methods such as IQR-LSTM, which this work employs. Simpler methods are generally easier to implement but require manual tuning and may struggle with complex temporal patterns, whereas ML-based methods can capture those patterns at the cost of greater data requirements and reduced interpretability. In

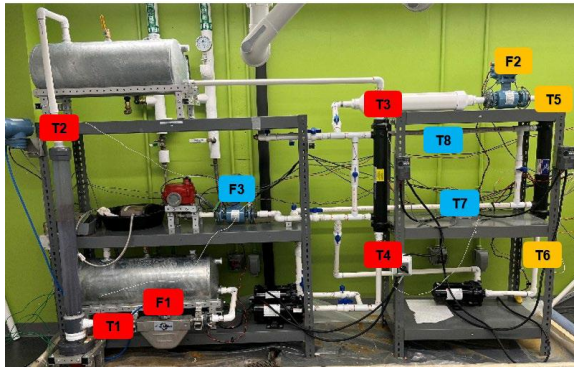
safety-critical applications such as nuclear systems, the consequences of both false positives and false negatives must be carefully considered (Ma & Jiang, 2011). Consequently, selecting an anomaly detection method requires balancing detection performance, interpretability, computational complexity, and application-specific constraints.

Anomaly detection is a vast research area in its own right, and numerous techniques could be adapted for this application. While the field continues to evolve, the objective of this work is not to develop or benchmark new anomaly detection algorithms. Instead, the focus is on integrating a representative anomaly-detection method into a complete DT framework to demonstrate the DT's forecasting and fault-detection capabilities. Accordingly, the IQR-LSTM approach is adopted as a representative data-driven method, while optimization and comparative evaluation of anomaly detection algorithms are reserved for future work.

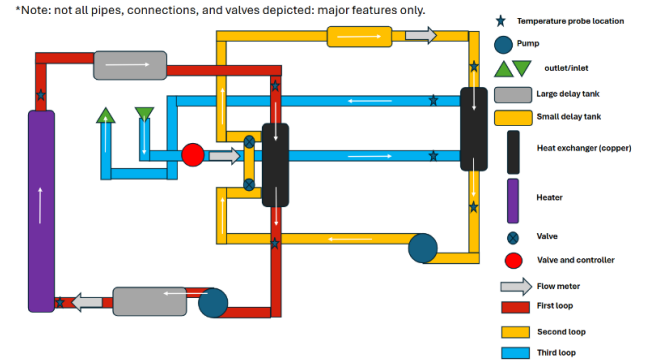
3. METHODOLOGY

3.1. Experimental System

The Tri-Twin, located at the University of Tennessee, Knoxville (UTK), is a three-loop water system that serves as a test loop to advance digital engineering, particularly in anomaly detection and DT development. The system is designed with features for digital engineering, system measurements, anomaly detection, and digital control. Figure 4(A) shows the physical Tri-Twin system with labeled sensor positions, and Figure 4(B) provides a simplified diagram of the components and sensors within the loops. This system has been used for various digital engineering developments and continues to serve as a test bed for ongoing research and development (Carnal et al., 2025).



(A)



(B)

Figure 4. (A) The Tri-Twin system with labeled temperature and flow meter locations; (B) the illustrative system diagram (Carnal et al., 2025)

The Tri-Twin consists of three physically separated loops. Loop one, or the primary loop, includes a heater, two delay tanks, a pump, and a heat exchanger that connects it to the intermediate loop. Loop two, the intermediate loop, contains

a pump and two heat exchangers, one connecting it to the primary loop and the other to loop three. Finally, loop three, or the balance-of-plant loop, is connected to the intermediate loop via the second heat exchanger. The balance-of-plant

loop does not have a pump, but the flow rate is actively controlled by a combination of building water pressure and the motor-operated valve (MOV) position. The primary loop has four temperature probes, while the intermediate and balance-of-plant loops each have two. Each loop has a flow meter measuring the flow rate in gal/min. The loops are controlled via LabVIEW, with all sensors connected to an NI-based data acquisition system (DAQ).

The primary loop is the heated loop. In an energy system, this would be the reactor or the heat-producing component. Pump one controls the flow in this loop; a limit was set to ensure the flow command never drops below 0.5V, maintaining a minimum flow of ~1 gal/min. The pump's maximum speed is achieved with a 5V command signal, corresponding to approximately 15 gal/min. The heater can be set to any voltage between 0V and 5V, with 0V corresponding to the heater being off and 5V corresponding to ~9,000 watts. The heat exchanger connecting loop one and loop two is a shell-and-tube heat exchanger. Loop two serves as an intermediate loop, with the primary function of removing heat from the primary loop via the heat exchanger. This loop's flow is controlled by pump two and is designed to operate between 0.5V and 5V, with corresponding flow rates similar to those of the primary loop. Finally, the balance of the plant loop, connected to the intermediate loop via another shell-and-tube heat exchanger, serves as the heat sink. This is where heat is removed from the system; as in an energy system, the turbines and their respective cooling loops act as the heat sinks. The MOV in the balance of the plant loop is connected in proportion to the heater in the primary loop. This means that if the heater command is set to 2V (~6,000 W), the MOV command is set to 2V (valve is 40% open); if the heater is at 5V (~9,000 W), the MOV is fully open (at 5V). This creates a direct relationship between the heat source and the heat sink flow, conceptually mimicking the relationship seen in larger energy plants.

3.1.1. Data Collection

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) was used for system control, data acquisition, and signal processing. It provides a graphical programming environment for implementing control and measurement applications. For sensor connections, National Instruments (NI) modules served as hardware components and were configured via the LabVIEW interface. Through the NI modules, connections were made to the three flow meters (one in each loop), eight temperature probes, and the pumps, heater, and valve, which served as inputs/actuators to control the loop. This setup enabled easy integration with the virtual environment. Using LabVIEW, inputs, flow, and temperatures were collected on demand and appended to a single CSV file. When the LabVIEW setup ran, data collection began, and the operator could control the heater, pump, and valve voltages. The current setup collected a data point approximately every four seconds. Collected data

included steady-state, transient, and anomaly-injection runs. These runs were used throughout this analysis; their parameters and methods will be discussed later in the paper, as applicable. During data collection, the operator could adjust the inputs as desired; all adjustments were made via the LabVIEW VI, shown in Figure 5.

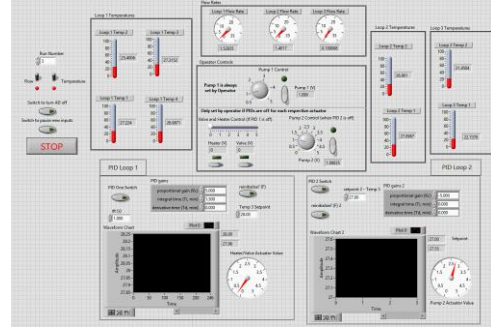


Figure 5. Finalized LabVIEW VI for the Tri-Twin

3.1.2. PID Controls

Proportional-integral-derivative (PID) control is employed as the control scheme for this system. This was done using two separate PIDs, one for the primary loop and one for the intermediate loop. For the primary loop, the PID used the heater as the actuator, with the heat exchanger inlet temperature as the setpoint (temperature probe three). For the second PID, the pump in the intermediate loop is the actuator, and the process variable/setpoint is the hot-side temperature of the heat exchanger in the intermediate loop (temperature probe five). Figure 6 shows the PID scheme for the primary loop, and Figure 7 shows it for the intermediate loop. This PID scheme was chosen to vaguely represent that of a reactor-like system, in which the actuators work to keep the system's temperatures within a certain setpoint.

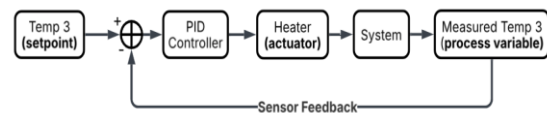


Figure 6. Primary loop PID control scheme (PID 1)

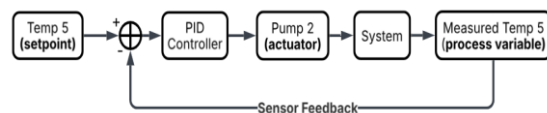


Figure 7. Intermediate loop PID control scheme (PID 2)

The PID gains were tuned manually to achieve the desired performance in terms of rise and settling time. The currently optimal gain and time values are shown in Table 1 below. The proportional gain of PID 2 is negative because it is a reverse-acting control loop. This means that as the pump voltage increases, the hot-side temperature decreases. In industry, PID control is usually implemented by selecting the reverse

operation and using a positive proportional gain multiplier. With that, the negative gain here is just a convention of the control action, but the idea and tuning approach are similar to those seen in industrial control.

Table 1. Tuned gains for each PID

	PID 1	PID 2
Proportional Gain	10	-3.5
Integral Time (min)	2	7.5
Derivative Time (min)	0	0

3.2. LSTM Implementation

The decision to use a multi-output architecture and an LSTM was based on an extensive sensitivity analysis presented in past work (Bailey et al., 2025). Here, we explored the performance of RNNs, GRUs, and LSTMs, as well as a single model for all features (multi-output) versus a separate model for each feature (single-output). It was determined that LSTM performed best, but both RNN and GRU still performed relatively well. The LSTM outperformed both other time-series models for 8 of the 11 features, making it a clear choice. The one disadvantage of the LSTM is that it is a larger, more complex model that can influence training time, but due to the relatively small size of the data intake for this system, it did not result in a significant time difference that would influence the reason to go with the RNN or GRU. Additionally, performance was fairly even between the single-output models and the multi-output model, with the single-output model providing slightly better results, as each model was optimized specifically for each individual feature. However, in terms of implementation, it is not realistic to use 11 separate models to predict each feature when a single model can perform nearly as well, and it makes the integration more seamless for this DT approach.

In prior work, a novel method, referred to as correlation-informed LSTM (CI-LSTM), was developed to improve predictions by leveraging correlations in the data using a combined LSTM-linear regression loss function. This method is not explored in depth in this work, but it offers a path for future implementation for improved predictions (Bailey et al., 2026).

In this work, two data-splitting approaches were used for model training and implementation. The first used a standard 70% training, 10% validation, and 20% testing split. The data order was preserved throughout the split to ensure that no future information was available prior to prediction. However, this approach limits the model to predicting conditions represented in the training data. As a result, predictions may become unreliable when the system encounters transient behavior or operating conditions not observed during training. The second approach used a stacked dataset structure to improve model generalization across different operating conditions and enable prediction under varying parameter settings. For this approach, 18

complete datasets were generated, each containing more than 1000 s of data collected under different system parameters and including both transient and steady-state operation. 16 datasets were used for training, one for validation and parameter optimization, and one for testing. The datasets remained separate and retained their original temporal ordering, but were stacked and flattened along the sample axis to create a single training dataset. Rather than predicting absolute sensor values, the model was trained on the first differences of the target variables (step-to-step changes). This promotes stationarity and allows the model to better generalize across datasets with different baseline conditions. During inference, absolute predictions were reconstructed by accumulating the predicted changes onto the last known measurement. The training datasets covered the full operable range of pump, valve, and heater voltages, as well as a variety of transient inputs. This allowed the model to learn not only the relationships between the measured features under different operating conditions, but also the effects of the operating parameters themselves on system behavior. As a result, the LSTM was able to predict system features across the range of Tri-Twin operating conditions. This method, while more advanced in training and model setup, is better suited to online monitoring systems, particularly those with varying run parameters. Figure 8 shows the distribution of actuator parameters across the training datasets, illustrating the spread across the operating parameters.

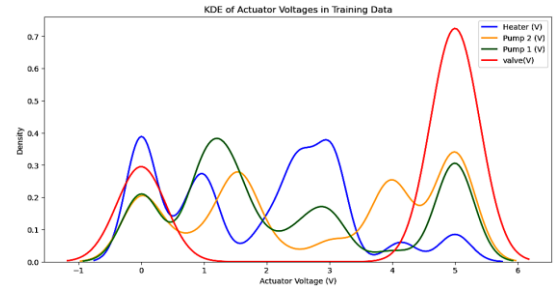


Figure 8. Distribution of actuator voltages in training data

An important consideration when working with time-series-based NN techniques is the look-back window, also called the sequence length, which determines how many past points the model can use to predict the next step. There is also the look-ahead horizon, which is the number of points the model predicts into the future at each step. Both are parameters that are optimized when building these models (Lipton et al., 2015). The look-back window size determines how many prior time steps the predictor has access to. The default is one time step, but it can be adjusted if a more complex system-memory model is required (MATLAB & Simulink, n.d.). Additionally, the look-ahead window can be adjusted, allowing you to specify how many future time steps to predict at each step. Once again, the default value is one, but it can be modified to predict multiple steps.

There are multiple ways to approach the forecasting horizon for LSTM modeling. These are illustrated in Figure 9 below,

which shows three of the most common options: one-step-ahead predictions, a fixed horizon, and a multi-horizon. One-step-ahead prediction involves predicting $n+1$, where n is the last available set of features before predictions are started. In this case, if continuous predictions are to be made, you would use a sliding window: $n+1$ is predicted, then the window slides forward one step, containing the number of features used for the look-back horizon, and the next prediction one step ahead is made. To increase the prediction horizon, either a fixed horizon or a multi-horizon can be used. For a fixed horizon, only one prediction is made, but it is for a step far in advance rather than just one step ahead. Finally, the multi-step horizon is the set of time steps $n+1$ through $n+h$, where h is the desired horizon length. For the offline LSTM in this work, one-step-ahead prediction was used for the model setup and to explore different look-back horizons. The multi-horizon technique was also used for online monitoring to allow for predictions farther into the future. Another consideration here is whether the lookback window is filled with the measured data or previous predictions. In this work, predictions are used unless the model error exceeds a set threshold; then it uses measured data to stabilize again before returning to predictions.

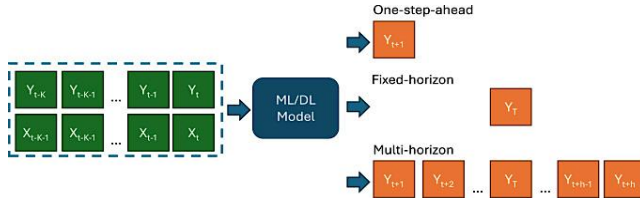


Figure 9. Forecast horizon approaches for LSTM predictions (Martín-Suazo et al., 2024)

Model performance was evaluated using root mean square error (RMSE) and relative root mean square error (rRMSE). Equation (1) illustrates the calculation of RMSE. This value has the same units as the feature and uses both the actual and predicted values to determine the error. The rRMSE is expressed as a percentage, indicating the error, as shown in Equation (2). In both equations, N is the number of data points, y is the actual value, and $\hat{y}$ is the predicted value. RMSE expresses prediction error in the feature's engineering units, providing an intuitive measure of model accuracy, whereas rRMSE expresses the error relative to the feature's magnitude. This normalization enables direct comparison of predictive performance across features with different engineering units and scales.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (1)$$

$$rRMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N \left(\frac{y_i - \hat{y}_i}{y_i} \right)^2} \times 100\% \quad (2)$$

3.3. Anomaly Detection

Due to the operational nature of the system, which functions under varying conditions and frequent transient behavior, non-adaptive anomaly detection techniques were found to be insufficient. With this, methods capable of adapting to the data's changing statistical properties were focused on. During offline testing, several approaches were evaluated, including LSTM-based thresholding, SPRT, and IQR-LSTM. Among these, the IQR-LSTM hybrid method demonstrated the most consistent and reliable performance and was therefore selected for further development and integration. IQR-LSTM is a method inspired by recent LSTM- and decision tree-based approaches, providing a way to handle dynamic systems without misclassifying intentional transients (Saha et al., 2024). The IQR-LSTM method combines time-series prediction with statistical outlier detection. The LSTM model captures temporal dependencies and generates predictions; anomaly detection is subsequently performed on the prediction residuals. Unlike static thresholding methods, this approach is more robust to transient fluctuations, reducing false positives. The IQR is computed from the residuals by identifying the first quartile (Q1) and third quartile (Q3), corresponding to the 25th and 75th percentiles. The IQR is defined as the difference between Q3 and Q1. Lower and upper bounds for anomaly detection are then established using Eqs. (3) and (4), where k is a tunable parameter that controls detection sensitivity. Residuals falling outside these bounds are flagged as anomalies. Figure 10 illustrates the framework of the IQR-LSTM method and its application to anomaly detection.

$$Lower\ Bound = Q1 - k \times IQR \quad (3)$$

$$Upper\ Bound = Q3 + k \times IQR \quad (4)$$

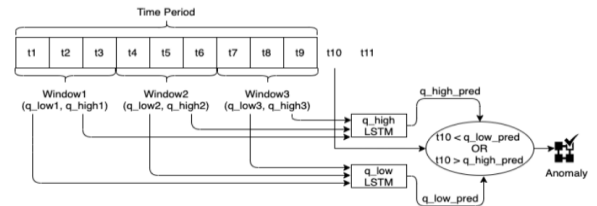


Figure 10. Overview of IQR-LSTM (Saha et al., 2024)

For implementing IQR-LSTM for this DT, the LSTM model is first used to capture temporal dependencies and generate predictions. Anomaly detection is then performed on the prediction residuals. The first (Q1) and third (Q3) quartiles of the residuals are computed over a rolling window. The value of k was determined empirically through offline experimentation by evaluating model performance on both nominal and abnormal datasets. This process involved iteratively adjusting k to balance minimizing false positives and false negatives. While this manual tuning approach provided satisfactory results, developing a more robust optimization strategy remains an important area for future

work. The parameters obtained during offline tuning were used as the initial configuration for the online anomaly detection system and were further refined based on real-time performance. To address non-stationary behavior and system drift, the IQR is computed over a rolling window, allowing detection thresholds to adapt dynamically to evolving data distributions. In addition, temporal persistence filtering is applied to ensure that anomalies persist for a specified duration, thereby reducing sensitivity to noise and short-lived fluctuations. As for detection itself, a two-road framework was employed, consisting of a system-level detector and an individual sensor-level detector. The system-level detector uses a multi-sensor voting mechanism that requires a minimum number of concurrent sensor anomalies to classify a system-wide event. This approach reduces false positives caused by isolated sensor noise. In parallel, a higher thresholding scheme is applied at the individual sensor level to identify extreme deviations.

For example, consider a physical bypass of heat exchanger 1 in which no water is lost, but a portion of the flow bypasses the heat exchanger and rejoins the system downstream. This physical change does not alter the control commands received by the DT. Consequently, the LSTM continues generating predictions based on the commanded operating state, while the measured data reflect the altered physical configuration. The resulting discrepancy between the measured and predicted temperatures increases the prediction residuals. This response is expected to appear most prominently at temperature probe 4 and subsequently affect downstream temperature measurements in loop 1 and, through system interactions, loop 2. Once the residuals exceed the adaptive IQR-based bounds and satisfy the temporal-persistence criteria, the affected sensor-level detectors flag the deviations as anomalies. If the required number of sensors produces concurrent anomaly flags, the system-level voting mechanism classifies the behavior as a system-wide event. Although the detection thresholds and sensitivity remain under evaluation, this example illustrates the expected sequence by which the proposed method detects a physical change not represented by the system's control commands.

3.4. DT Construction

To construct a true, complete DT, each discussed aspect must be interfaced, along with a communicator from the digital system back to the physical system (Lussier et al., 2025). The full integration of the DT focused on establishing bidirectional communication to ensure that LabVIEW, Python, and the data acquisition system worked together and responded appropriately. For this purpose, threading was used in the master Python script so that only one script handled the LabVIEW connection. This master script calls the necessary functions and controls data analysis, predictions, anomaly detection, and operator notifications. With a successful online prediction code and real-time

anomaly detection in place, the final step is to add an operator notification to complete the DT architecture. This was achieved by displaying a simple GUI when an anomaly was detected. This GUI informs the operator that an anomaly has been detected and that the operator should respond accordingly. This connection assumes the operator is the final informed decision-maker who can effectively make a decision based on the information provided and act accordingly. This is a *human-in-the-loop* DT setup, in which the operator is considered part of the physical system, and insights provided to the operator, enabling more informed decision-making, help complete the digital-to-physical interfacing. Figure 11 shows the flow of this DT setup.

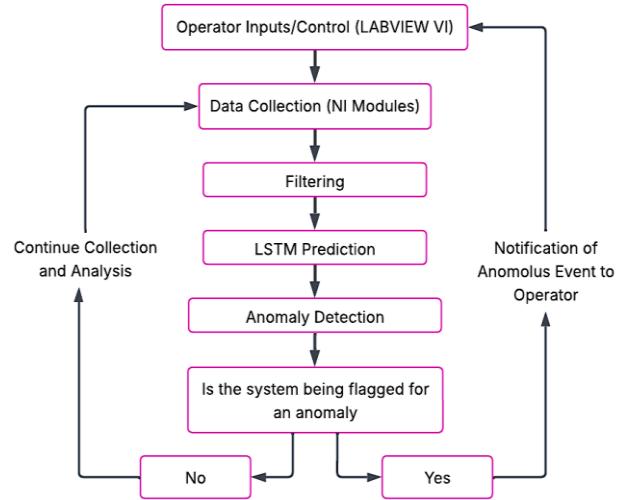


Figure 11. Digital system flow diagram

The DT for this test bed uses LabVIEW for data acquisition and control. However, industrial implementation could integrate the framework with a distributed control system (DCS) via a suitable industrial communication architecture. For example, an OPC UA client could exchange data with the DCS, while an industrial Internet of Things (IIoT) gateway could translate the data and publish it via an MQTT broker to communicate with the DT. Bidirectional communication could also allow validated, operator-approved commands to propagate in the reverse direction. Such an implementation would require further consideration of system integration, cybersecurity, access control, validation, and safety interlocks, but would offer a pathway toward a more representative architecture for plant operations (Ndum et al., 2026). While not explored in this paper, this is a major consideration for future work.

4. RESULTS

4.1. DT Framework

Figure 12 depicts the framework for the complete DT developed in this work. This shows a fully integrated DT, in which the physical system streams data to the digital system.

The digital system comprises a LabVIEW interface and Python nodal connections. This enables real-time analysis of incoming data, providing predictions and plotting comparisons between measured and predicted data while the anomaly detection methodology runs in the background. If an anomaly is detected, the system operator is notified via an

operator notification displayed in a Python GUI popup. This integrates the digital and physical systems through the operator's notification. This enables the physical system to derive insights from the digital system's analysis and to make adjustments accordingly.

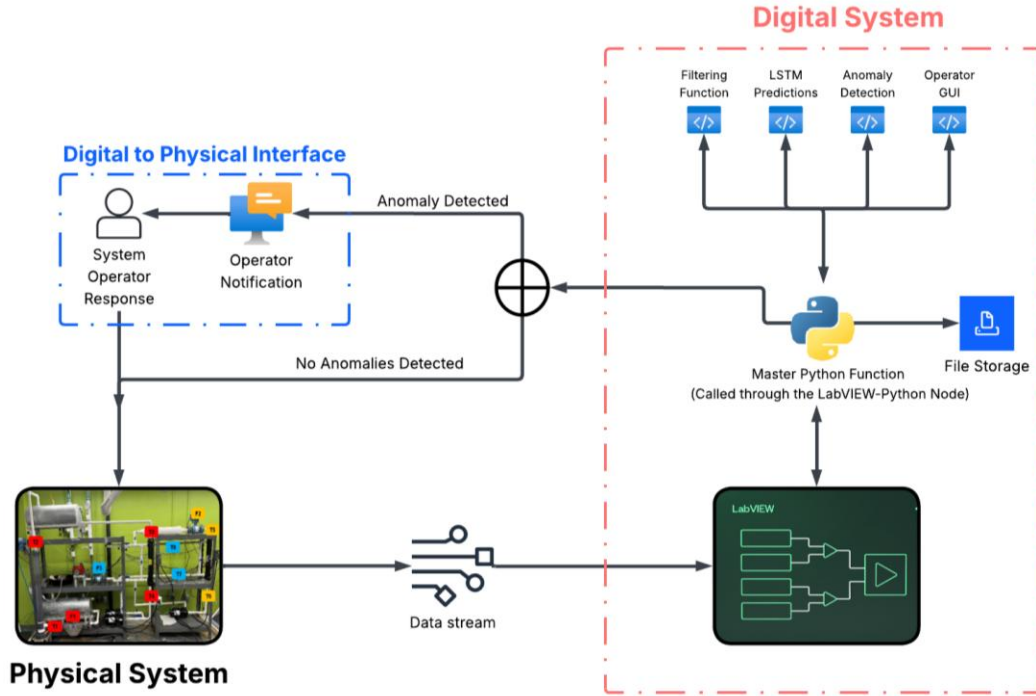


Figure 12. Illustration of the complete integrated DT for the Tri-Twin system

This complete DT framework offers a fully integrated approach to online monitoring and system operations for the flow loop. Because it uses standard industry software, such as LabVIEW, Python, and NI modules, this framework can be adapted for a range of applications. This accessible format allows much of the analysis to be conducted in Python files and provides a clear, simple connection, ensuring efficient communication between the physical and digital systems. This complete integration of digital and physical systems provides a clear framework for DT development of experimental systems and offers a fairly modular setup that will enable further development and exploration in future work.

4.2. Offline Feature Predictions

To initially build the models for the complete DT, the LSTM was trained and optimized offline. The first model was trained using the standard procedure, with the first 70% of the data for training, 10% for validation, and the remaining 20% for testing. Figures 13 and 14 present the resulting flow and temperature profiles, respectively, for an experimental run with both pumps at 3V and the heater at 5V. The shaded regions around the results represent the errors reported by the flow meter and temperature sensor probe manufacturers (DWYEROMEGA, n.d.; Emerson Automation Solutions,

2020). Table 2 displays the resulting RMSE and rRMSE values. Overall, the model performs well on the final 20% of the dataset, achieving high predictive accuracy. However, high performance with the LSTM model using the traditional data-splitting technique means the model is accurate only when the system is run under the same parameters used during training.

Table 2. LSTM performance using the traditional data splitting approach

	RMSE	rRMSE
	Gal/Min	%
Flow 1	0.031	0.46
Flow 2	0.025	0.28
Flow 3	0.021	4.93
	°C	%
Temp 1	0.047	0.21
Temp 2	0.114	0.49
Temp 3	0.044	0.19
Temp 4	0.087	0.38
Temp 5	0.052	0.23
Temp 6	0.031	0.13
Temp 7	0.020	0.10
Temp 8	0.026	0.13

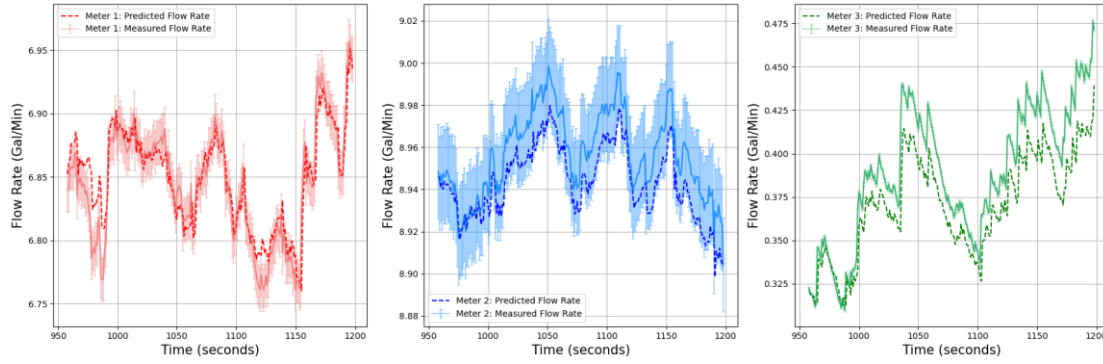


Figure 13. Flow predictions for the single dataset splitting training method

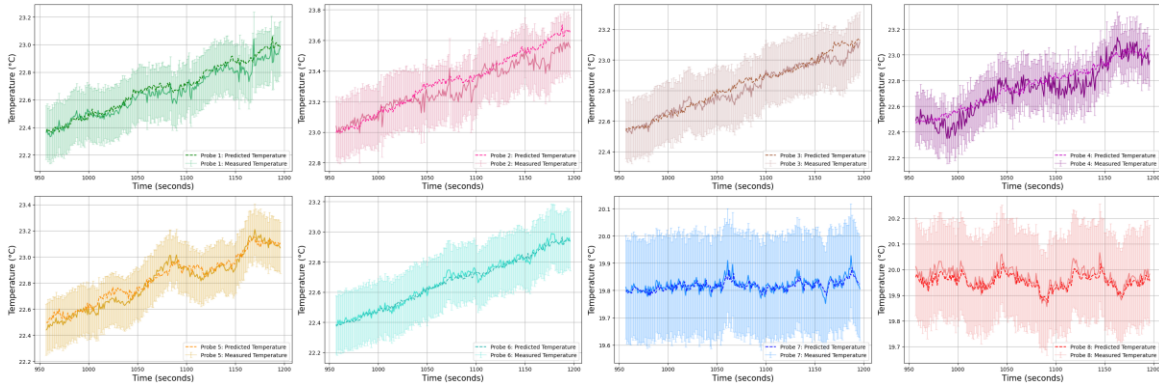


Figure 14. Temperature predictions for the single dataset splitting training method

To further explore the best LSTM model for online monitoring, a sensitivity analysis was conducted across different lookback horizons. This was done only for the stacked training method, since a sensitivity analysis of the traditional data-splitting method was presented in prior work

and the stacked method is more operationally relevant to online monitoring applications (Bailey et al., 2025). Table 3 presents the RMSE and rRMSE results using lookback horizons of 1, 15, 30, and 60 data points¹.

Table 3. Look-back horizon sensitivity analysis¹

	Look-Back Value (Look Forward=1)							
	1		15		30		60	
	RMSE	rRMSE	RMSE	rRMSE	RMSE	rRMSE	RMSE	rRMSE
	Gal/Min	%	Gal/Min	%	Gal/Min	%	Gal/Min	%
Flow 1	0.210	49.40	0.240	2.32	0.149	1.48	0.244	2.23
Flow 2	0.154	53.66	0.149	9.67	0.108	7.30	0.097	4.36
Flow 3	0.056	4.99	0.050	4.46	0.055	5.02	0.048	4.45
	°C	%	°C	%	°C	%	°C	%
Temp 1	0.254	1.06	0.187	0.766	0.142	0.59	0.151	0.62
Temp 2	0.437	1.83	0.382	1.60	0.594	2.46	0.317	1.31
Temp 3	0.065	0.26	0.047	0.19	0.067	0.27	0.065	0.27
Temp 4	0.162	0.67	0.199	0.82	0.145	0.60	0.169	0.70
Temp 5	0.124	0.52	0.156	0.66	0.131	0.55	0.216	0.92
Temp 6	0.206	1.03	0.218	1.01	0.135	0.64	0.170	0.79
Temp 7	0.277	2.00	0.246	1.77	0.200	1.44	0.220	1.58
Temp 8	0.169	0.89	0.146	0.72	0.133	0.66	0.147	0.71

¹The highlighted cells indicate the best result for each feature

For the results in Table 3, a look-forward horizon of 1 was used, meaning the model predicts only 1 value ahead. For each lookback value, Optuna² was used to optimize the hyperparameters, and five trials were run. Optuna is an automatic hyperparameter optimization framework that enables efficient exploration of the desired hyperparameter search spaces. The number of trials is the number of complete executions of the objective function, with each trial attempting different hyperparameter values. The number of trials in this work could be increased, but because this is a more general analysis of performance across different look-

back values, five runs were sufficient. With a lookback value of 1, it took 2.2 minutes to run through the trials; with a value of 60, it took 46.1 minutes. This is a significant difference that highlights the need to balance accuracy and optimization time. Accordingly, the 30-point lookback horizon was deemed optimal and used for all LSTM models. Figures 15 and 16 show how the model trained using the stacked technique performed on the test data for flow-rate and temperature predictions, respectively, using the look-back value of 30. The quantitative results of these plots are shown in Table 3 in the 30-look-back-value column.

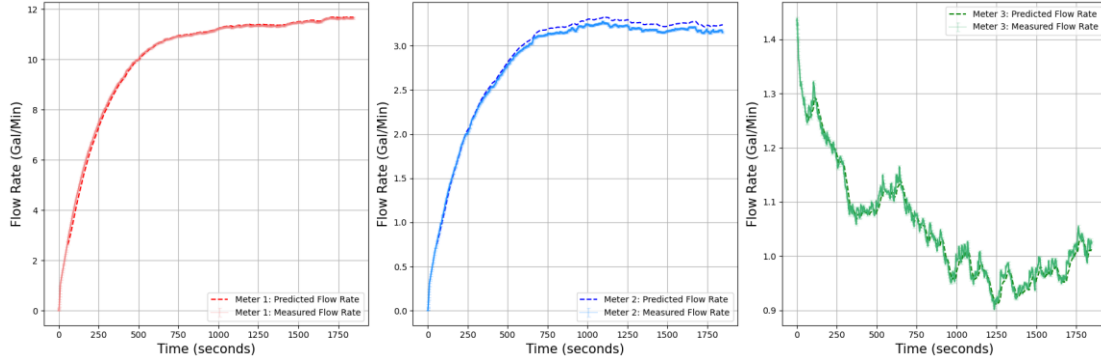


Figure 15. Flow predictions based on the stacked training method with a lookback window of 30

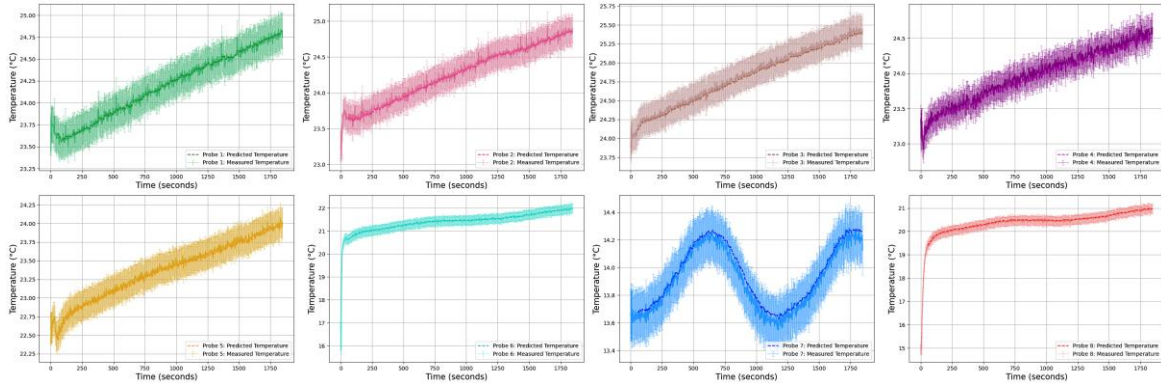


Figure 16. Temperature predictions based on the stacked training method with a lookback window of 30

4.3. Online Monitoring

For online predictions, the LSTM model forecasts 10 seconds ahead. As a new data point is collected, it uses those inputs to predict the next 10 seconds: 9 of those seconds have their predictions updated, and the 10th second is a new prediction. This method prevents an exponential increase in the number of predictions; instead, it always predicts 10 seconds ahead. In LabVIEW, a Python node was used to call Python functions for data filtering, model predictions (LSTM model), anomaly detection (IQR-LSTM), real-time plotting, and notifications. Instead of a separate Python node for each function, all functions were organized to be called from a

master file. This file was then connected to LabVIEW. The master Python script invoked each function as needed and used threading to manage file I/O. This enabled smooth transitions, depending on which file needed to be opened for each function. This approach also enabled real-time connectivity. Once the LabVIEW function executed, it invoked the Python file and initiated data filtering. Once the filtered data reached the length of the LSTM's lookback window, the LSTM would begin making predictions, and the plotting GUI would plot both the filtered measured data and the predicted values as they are collected and predicted, respectively. In the backend, the anomaly detection script evaluates the model's outputs to determine whether the

² <https://github.com/optuna/optuna.git>

system is in a nominal or anomalous state. There is some lag in this approach, largely because file access can be delayed if one function attempts to open a file and is blocked by another function that is reading or writing it. This simplification in data transfer has been shown to be sufficient given this level of data collection; however, more advanced data transfer techniques could be explored in future work to improve its

applicability to more complex systems. With everything set up for online monitoring, Figure 17 shows a snapshot of what the operator sees as data is being collected. The operator can see the overall RMSE and the windowed RMSE, which is the RMSE for the most recent 20 collection points for each feature. Lastly, the shaded regions are the manufacturer's predicted error bars as discussed earlier.

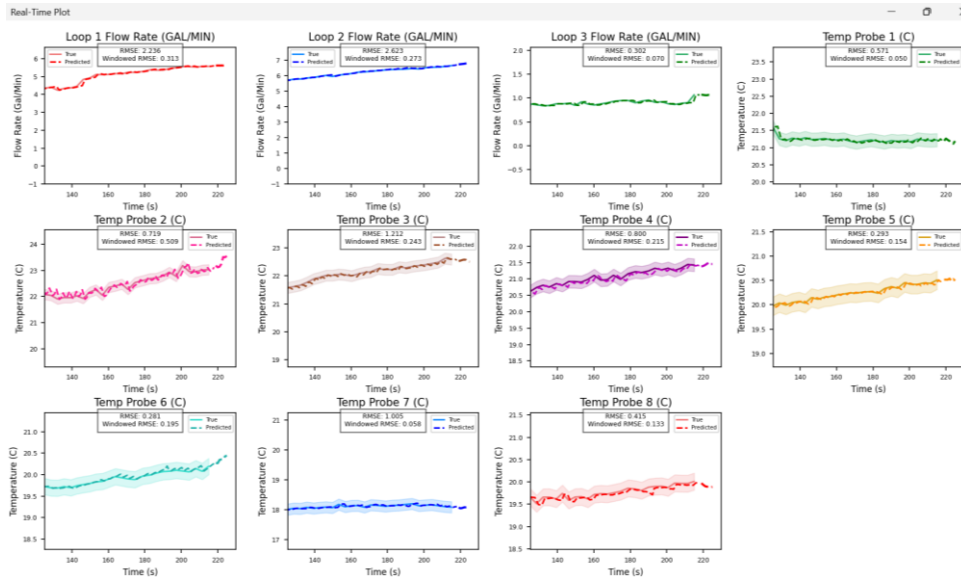


Figure 17. System measurements and a real-time prediction plot that the operator sees during operations

4.4. Digital-to-Physical Communication

Regarding operator pop-up notifications, the first one an operator can encounter is shown in Figure 18. In the LabVIEW interface, the operator can disable anomaly detection. This might be done if the operator plans to run specific runs that may involve consistent transients, or if changes to the system or models are being made. While this is possible, the operator must exercise extreme caution when operating the system with anomaly detection turned off.

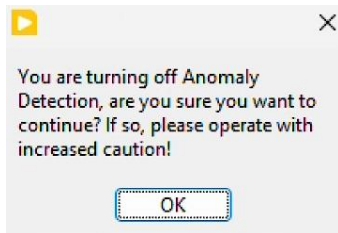


Figure 18. The notification if the operator turns off the anomaly detection

If an individual-feature anomaly is detected, a pop-up like the one shown in Figure 19 will appear. The pop-up displays the time the anomaly was detected, which feature or sensor triggered it, and a warning to monitor the system closely. The

operator can close the pop-up by clicking 'OK' or the close button. This notification is designed to provide the operator with insights into where the anomaly is occurring and to determine whether the system needs to be shut down or can continue operating.

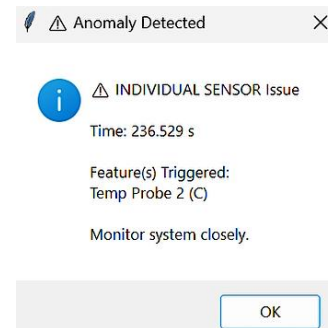


Figure 19. Individual sensor flag notification example

A similar notification appears when a system-level anomaly is detected, as shown in Figure 20. It displays the sensors that triggered it, indicates the likely location, and directs the operator to additional information. Currently, the locator only determines whether the anomaly is likely around the first heat exchanger (loops one and/or two) or around the second heat exchanger (loops two and/or three). During operation, the LabVIEW VI continuously plots the measured and predicted

values via the GUI. Anomalies trigger a notification above the plotting display, as shown in Figure 21, allowing the operator to respond immediately.

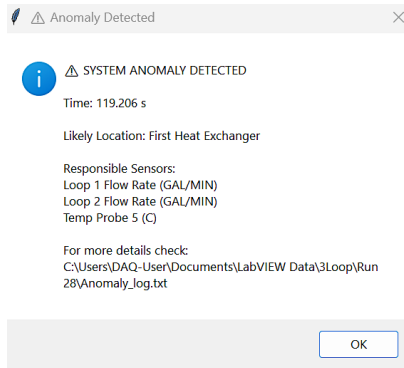


Figure 20. System-level anomaly flag notification example

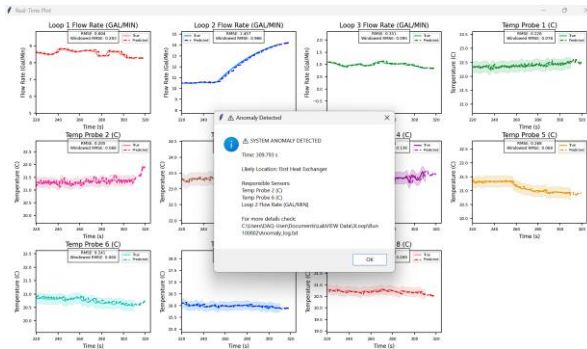


Figure 21. Operator display during an anomaly notification

5. CONCLUSION AND FUTURE WORK

This research presents the complete implementation of a DT framework for a forced-flow system. The LSTM network achieved high accuracy in feature prediction, while a fully threaded Python setup integrated into LabVIEW enabled real-time prediction and operator notifications. The resulting DT is functional, flexible, and modular, enabling the incorporation of future models and capabilities. The framework may also be adapted to related experimental systems to improve operations and operator insight. Key directions for future work include integrating a physics-based model, developing component-based DTs, and expanding automation through DT feedback.

The integration of a physics-based model for this system is under investigation, motivated by its potential to simulate experimental plans in advance and provide additional predictions that complement the ML approach. A system-level physics model of the Tri-Twin is being developed and validated (Barr et al., 2026) and could be incorporated into the framework presented here. Evaluating the physics-based and ML models independently would support the development of an effective hybrid approach. Another possibility is a physics-informed LSTM, in which known physical behavior is incorporated into the loss function.

Future work could also involve developing a component-based DT, which focuses on higher-fidelity models of each component rather than a lower-fidelity system-level model (Bowman et al., 2022). One ML-based approach would use a graph neural network (GNN) that leverages spatial relationships and component connectivity to improve predictions. Such models could provide more detailed insight into component degradation, detect localized changes, and simplify diagnostics. Lastly, another major area for expansion is autonomous control. An initial step could allow operators to shut down the system directly through a notification displayed on the HMI. This approach would support rapid intervention while keeping the final decision with the operator. With sufficient validation, the DT could respond autonomously to anomalies by shutting down the system or adjusting its operation to restore acceptable conditions. This builds on the current framework but requires more rigorous quantification of model performance and uncertainty to ensure trustworthiness.

Overall, the fully integrated DT framework presented here provides a successful proof of concept for system-level DT development while enhancing the monitoring and operation of the experimental facility.

ACKNOWLEDGEMENTS

This material is based upon work partially supported by the Department of Energy National Nuclear Security Administration through Defense Nuclear Nonproliferation's Enabling Capabilities in Technology Consortium under Award Number DE-NA0004197. This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or limited, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors herein do not necessarily state or reflect those of the United States Government or any agency thereof.

The authors acknowledge the use of OpenAI's GPT-5 to assist with coding, including Python-LabVIEW integration, debugging, code optimization, and syntax support. The responsibility for all content and results remains solely with the authors.

Finally, the authors would like to thank fellow graduate researchers Mattia Zanotelli and James Carnal, as well as undergraduate researchers Reagan Barr and Nicholas Armstrong, for their assistance with aspects of this work.

REFERENCES

- AIAA Digital Engineering Integration Committee. (2020). *DIGITAL TWIN: DEFINITION & VALUE | An AIAA and AIA Position Paper*. American Institute of Aeronautics and Astronautics, Inc.
- Attaran, M., & Celik, B. G. (2023). Digital Twin: Benefits, use cases, challenges, and opportunities. *Decision Analytics Journal*, 6, 100165. <https://doi.org/10.1016/j.dajour.2023.100165>
- Bailey, L., Barr, R., Carnal, J., Clark, A., Copley, B., Petersen, J., Coble, J., & Zhao, X. (2025). *Machine Learning for Time-Series Forecasting with Quantile Regression and Anomaly Detection*. Transactions of the American Nuclear Society, 133 (Winter Meeting). <https://doi.org/https://doi.org/10.13182/T133-49266>
- Bailey, L., Carnal, J., Coble, J., & Zhao, X. (2026). Correlation-Informed Time-Series Forecasting and Anomaly Detection. *2026 IEEE Aerospace Conference*, 1–11. <https://doi.org/10.1109/AERO66936.2026.11519810>
- Bandala, M., Chard, P., Cockbain, N., Dunphy, D., Eaves, D., Hutchinson, D., Lee, D., Ma, X., Marshall, S., Murray, P., Parker, A., Stirzaker, P., Taylor, C. J., Zabalza, J., & Joyce, M. J. (2024). Digital twin challenges and opportunities for nuclear fuel manufacturing applications. *Nuclear Engineering and Design*, 420, 113013. <https://doi.org/10.1016/j.nucengdes.2024.113013>
- Barr, R., Bailey, L., & Zhao, X. (2026). *Development of a Modelica-Based Physics Model for Digital Twin Implementation of a Three-Loop Flow System*. ANS Student Conference.
- Barricelli, B. R., Casiraghi, E., & Fogli, D. (2019). A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications. *IEEE Access*, 7, 167653–167671. <https://doi.org/10.1109/ACCESS.2019.2953499>
- Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2), 157–166. <https://doi.org/10.1109/72.279181>
- Bowman, D., Dwyer, L., Levers, A., Patterson, E. A., Purdie, S., & Vikhorev, K. (2022). A Unified Approach to Digital Twin Architecture—Proof-of-Concept Activity in the Nuclear Sector. *IEEE Access*, 10, 44691–44709. <https://doi.org/10.1109/ACCESS.2022.3161626>
- Crespi, N., Drobot, A. T., & Minerva, R. (Eds.). (2023). *The Digital Twin*. Springer International Publishing. <https://doi.org/10.1007/978-3-031-21343-4>
- Dhee, K. (2023). *Understanding the evolution of GPT: From encoder decoder to llms*. Medium.
- Gregory, C., Rabah, S., Chapurlat, V., Burčiar, F., & Herchlová, M. (2026). Leveraging model-based systems and software engineering for digital twin engineering: Methods and digital thread opportunities. *Journal of Industrial Information Integration*, 50, 101023. <https://doi.org/10.1016/j.jii.2025.101023>
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Huang, Q., Zeng, W., Liu, J., Zhang, Z., Deng, J., Qiu, Z., Xu, L., Wei, Z., Lu, Q., Gong, L., Shi, C., & Zhong, X. (2025). Shaping the future of nuclear reactors with digital twins: Current developments and perspectives. *Applied Energy*, 402, 126922. <https://doi.org/10.1016/j.apenergy.2025.126922>
- Lussier, J., Hua, E. Y., Ledford, A., Thomas, D., Van Bossuyt, D. L., & Wooley, A. (2025). Differentiating Between Digital Twins and Control Systems for Complex Systems. *Journal of Computing and Information Science in Engineering*, 25(12), 120807. <https://doi.org/10.1115/1.4070329>
- Martín-Suazo, S., Morón-López, J., Vakaruk, S., Karamchandani, A., Aguilar, J. A. P., Mozo, A., Gómez-Canaval, S., Vinyals, M., & Ortiz, J. M. (2024). Deep learning methods for multi-horizon long-term forecasting of Harmful Algal Blooms. *Knowledge-Based Systems*, 301, 112279. <https://doi.org/10.1016/j.knosys.2024.112279>
- Mengyan, H., Xueyan, Z., Cuiting, P., Yixuan, Z., & Jun, Y. (2024). Current status of digital twin architecture and application in nuclear energy field. *Annals of Nuclear Energy*, 202, 110491. <https://doi.org/10.1016/j.anucene.2024.110491>
- Ndum, Z. N., Lim, D., Ford, J., Adu, S., Tao, J., Hassan, Y., & Liu, Y. (2026). Large language model-assisted digital twin for remote monitoring and control of advanced reactors. *Progress in Nuclear Energy*, 192, 106172. <https://doi.org/10.1016/j.pnucene.2025.106172>
- Nuclear Regulatory Commission (NRC). (2025). *Digital Twins*. Nuclear Regulatory Commission. <https://www.nrc.gov/reactors/power/digital-twins>
- Saha, S., Sarkar, J., Dhavala, S., Sarkar, S., & Mota, P. (2024). Quantile LSTM: A Robust LSTM for Anomaly Detection In Time Series Data. *IEEE Transactions on Artificial Intelligence*, 5(8), 3939–3950. <https://doi.org/10.1109/TAI.2024.3353163>
- Zio, E., & Miqueles, L. (2024). Digital twins in safety analysis, risk assessment and emergency management. *Reliability Engineering & System Safety*, 246, 110040. <https://doi.org/10.1016/j.res.2024.110040>