

Article

A Scalable AI-Driven Digital Twin Framework for Hydroelectric Power Plants

Israel Silber^{1,*} , Scott Warnick¹, Michael R. Kieburz¹ , Shuhao Bai¹, Sherman Beus¹, Matt C. Macduff¹, Greg Kenyon² and Chitra Sivaraman¹

¹ Pacific Northwest National Laboratory, Richland, WA 99352, USA

² Tacoma Public Utilities, Tacoma, WA 98409, USA

* Correspondence: israel.silber@pnnl.gov

Abstract

The growing U.S. energy demand, combined with the cost reduction and reliability enhancement needs of existing hydropower plants, necessitates smart modernization. Digital twin (DT) frameworks can enhance hydropower operations by leveraging advanced data-driven models for energy production optimization, degradation detection, anomaly prediction, and early-warning alarms. DTs can not only improve operational efficiency and asset longevity but also strengthen energy dominance in critical infrastructure by minimizing costly downtimes. Here, we present a new, scalable DT framework developed to support the aging fleet of energy-generating hydropower dams. This digital twin framework is built upon the application of a long short-term memory (LSTM) algorithm, enabling high-fidelity modeling and real-time monitoring of energy asset operations. By synthesizing preprocessed historical hydropower generating unit sensor and operational parameter data and using a model stacking approach, the DT achieves consistent accuracy in simulating multiple aspects of unit behavior, thereby enabling predictive insights. Following the description of the DT components, we demonstrate, using data collected at the Alder Dam in the Pacific Northwest, the framework application for hydropower unit characterization, showcase the DT's routine ("online") and simulation ("offline") modes, available on our operational data portal, which hosts data from multiple hydropower facilities, and discuss future capabilities and expansions.

Keywords: artificial intelligence; digital twins; hydroelectric power; long short-term memory (LSTM); machine learning; maintenance engineering; prediction methods; scalability



Academic Editors: Yong Liu and Juan I. Pérez-Díaz

Received: 29 July 2026

Revised: 19 August 2026

Accepted: 28 September 2026

Published: 2 October 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

The U.S. hydroelectric power plant fleet, consisting of more than 2000 facilities, is a crucial component of the country's electricity sector, accounting for approximately 6% of U.S. energy production capacity as of 2025 [1,2]. The bulk of the U.S. hydropower generation capacity is concentrated in the Pacific Northwest [2], which contains a mix of some of the largest plants in the country, in addition to mid- and small-capacity facilities. However, this hydropower fleet is aging, potentially causing an increase in operation and maintenance costs, which tend to be significantly higher at smaller facilities [2]. Even when capacity upgrades are considered, facility aging likely accounts for a significant fraction of the long-term deterioration of the U.S. hydropower fleetwide capacity, which has decreased by 13% since 1980 [3].

The aim to reduce operation and maintenance costs, combined with growing energy demand, necessitates optimization of existing hydropower assets, where reliability and resilience are paramount. Digital twin (DT) frameworks have the potential to address these aspects across multiple facets of hydropower operations [4–6]. The purposes of DTs aiming to serve the hydropower industry include pumped-storage hydropower (PSH) optimization [7,8], fault detection [9,10], and power generation and related governor controller predictions [11,12].

Different optimization techniques and solvers have previously been proposed and demonstrated for various DTs [8,11]. However, given the nonlinearity of such systems, artificial intelligence, specifically machine-learning techniques, offers the most robust path forward, as demonstrated by recent conceptual and applied studies [7,9,12–16].

Although different variants of DTs for hydropower have been proposed and developed, mainly over the past decade, references for interfaced, operational frameworks (e.g., [17]) are scarce. This is due to, among other reasons, the highly dynamic nature of such power generation systems and relevant data availability, resolution, and completeness (see [13] and references therein).

Although certain effective sampling rate limitations can hinder progress in some aspects of hydropower systems, such as excitation system dynamics, many of the measurement-related challenges and limitations can be rectified via the application of certain data engineering and machine-learning techniques. Here, we present an operational DT framework developed at Pacific Northwest National Laboratory (PNNL), which leverages some of these methods to mitigate the above-mentioned limitations. The framework has a modular and flexible backend coupled with a fully interfaced and easy to use frontend. Currently operated at the single-hydropower-generating-unit level, the framework enables unit data characterization and real-time monitoring of energy asset operations (“online” mode) and provides the option of conducting governor control and generated power response simulations (“offline mode”). The DT’s modular structure is designed with scalability in mind, manifested in straightforward customization to address various operations-related challenges in Francis and Kaplan-type units. The DT predictions are currently built upon a long short-term memory (LSTM) algorithm, enabling high-fidelity modeling.

In the following sections, we describe the general structure of our DT framework (Section 2), discuss data collection and preprocessing (Section 3), describe the DT’s LSTM model architecture, configuration and training (Section 4), and demonstrate the LSTM model prediction capabilities in error analysis using data collected for a Francis unit located at the Alder Dam in the Pacific Northwest (Section 5). We then showcase data characterization insights that can be derived using some of the trained LSTM models and demonstrate the DT’s two operation modes on our designated data portal (Section 6). A short summary, conclusions, and outlook are given in Section 7.

2. DT Framework Structure

The general structure of our DT framework is illustrated in the flowchart depicted in Figure 1. Real-time data from multiple sensors connected to specific units (measuring generated power, terminal voltage, shaft speed, penstock pressure, etc.) are continuously collected by hydropower facility operators. A data subset, representing nominal operation conditions, ideally spanning a period of a few years, is normalized and used to train a set of LSTM models (see [18]). Expert judgement of the training curves and objective error analysis are used to evaluate the LSTM models’ training, as demonstrated in Section 5. Once the evaluation is complete, the set of trained LSTM models is stacked into the framework for “online” real-time predictions, “offline” simulations, or other objective-specific system-characterization tasks. Each of the depicted processes is elaborated on in the following sections.

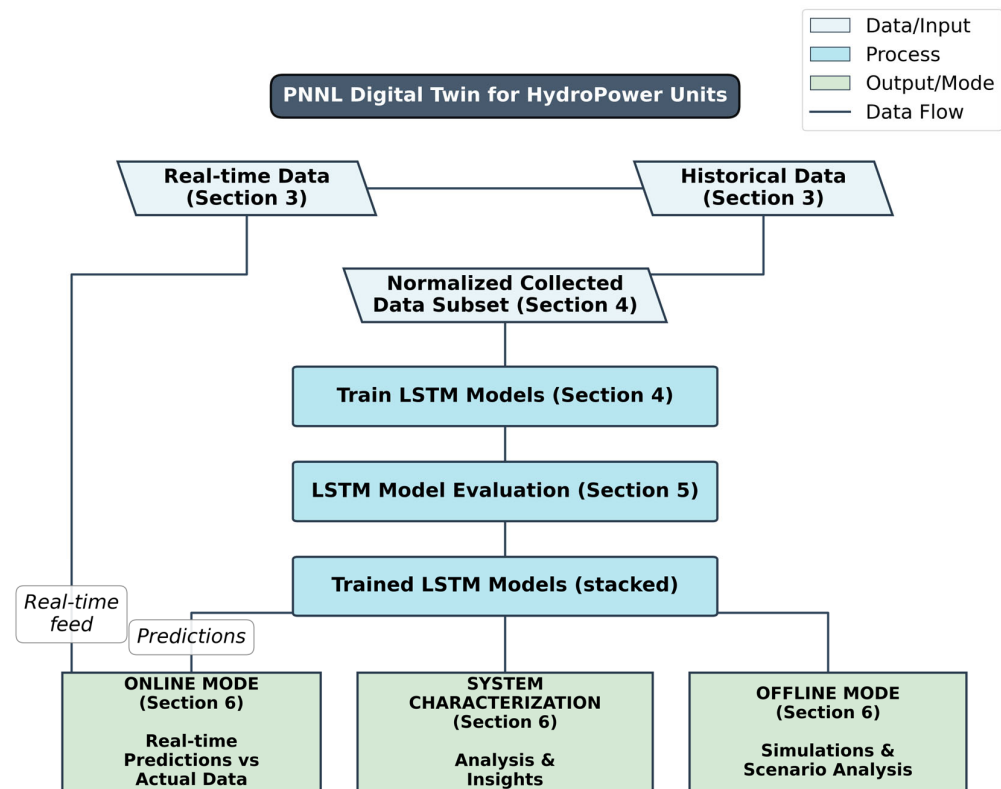


Figure 1. Long short-term memory (LSTM) model architecture diagram.

3. Data Acquisition and Preprocessing

Hydropower facilities typically collect operational data using various data logging software. Operator-specific collection configurations often lead to a nonuniform sampling rate across different parameters. In addition, some data logging software can occasionally record parameters at inconsistent intervals when the relative variability of these parameters does not exceed a certain threshold, resulting in apparent data trimming, most pronounced in “offline” mode simulations (see discussion in Section 4).

To standardize the data incorporated into the DT system, we developed and automated a data retrieval system that records relevant (available) facility parameters at a fixed 1 s interval, with linear interpolation in cases of longer sampling intervals or sporadic missing samples. This short 1 s interval can result in oversampling of some of the recorded parameters, such as the penstock pressure. However, as discussed next in Section 4, LSTM operation requires consistent sampling intervals without missing values. Although hydropower facilities typically collect data at effective sampling intervals ranging from 1 s to 1 min, depending on the logging software data acquisition configuration (sampling interval, deadband, etc.), certain potential DT objectives, such as accurately modeling generator frequency fluctuations or cavitation issues, require much higher sampling rates. Assuming site operators collect data at such high rates, our automated collection software and the subsequent workflow are adjustable to meet such objectives, as necessary.

4. LSTM Model Configuration and Training

4.1. Training Datasets

After the data are collected and formatted in standardized NetCDF files, subsets with minimal sampling gaps are normalized using the minimum and maximum properties per variable, which are based on observed and/or plausible value ranges. Those data subsets are then used as training data for the DT models. (This process of extracting and training

LSTM models is adaptable, allowing these steps to be tailored and repeated according to site operator needs and requests).

We generate two training datasets per unit: one for steady-state operation, during which the unit's breaker circuit is closed and the system is connected to the electric grid, and another for transition periods, specifically unit start-up and shut-down. In each dataset, periods corresponding to these operation statuses are subsetting, resulting in a dedicated training dataset with minimal periods of irrelevant data, rendering the training epochs more efficient. Because the transition periods are typically short and dispersed, we aim to maximize the training date span to gather as many transition instances as possible.

Because of the limited and/or inconsistent effective sampling rate of certain variables, attributable to the logging software configuration (as discussed above), model instability becomes more likely, especially in the case of recursive predictions. Variables that are more susceptible to an insufficient sampling rate, given their short-time-scale dynamics, typically, the shaft speed, terminal voltage, field voltage, field current, reactive power, and penstock pressure, are filtered using a Savitzky–Golay (SG) filter [19] with a window size of 60 s. This filtering procedure reduces sharp sample gradients stemming from a low and/or inconsistent effective sampling rate, thereby mitigating prediction errors and stabilizing the model's response to some extent.

4.2. LSTM Model Architecture and Setup

Our DT is built upon the LSTM algorithm provided in the Flax neural network library [20] of the JAX ecosystem [21]. An LSTM network is a type of recurrent neural network designed to learn and retain information over long sequences while addressing potential issues of standard RNNs, such as the vanishing gradient problem. Those attributes make LSTM algorithms a good fit for hydropower DTs, given the typical, temporally limited direct influence of parameter hysteresis on a given target variable's time series (orders of seconds or shorter). Other machine learning algorithms were considered, but were eventually not used because of typical hydropower unit data properties such as highly complex, non-linear sequences (excluding simpler algorithms such as Gated Recurrent Unit, e.g., [22]) or the general temporally limited direct influence of parameter hysteresis (excluding Transformer algorithms, for example, which may underperform in such cases, e.g., [22,23], unless heavily tuned and trained on very large datasets, beyond the records of some hydropower facilities).

We use a model stacking approach during training and for predicting target variables, primarily because complex system dynamics manifest in unique temporal patterns across the different target variables. Training an LSTM model to output multiple target variables with different patterns can skew the model weights (internal parameters) toward one variable over another, effectively necessitating a much larger number of weights to achieve satisfactory convergence.

Figure 2 shows a diagram describing the LSTM model architecture we employ. Considering the set of input variables (features; number represented by `num_features`), a given LSTM model requires its target variable(s) to be predicted (number represented by `num_targets`; in our case, `num_targets` equals 1). The model input layer receives a set of samples (the number of samples is determined by `batch_size`). Each sample is provided with a short history of its preceding lookback values, which are used to update the LSTM cell's hidden state (the cell's short-term memory; size of `hidden_state`). The LSTM cell output passes through a 64-neuron dense layer (a layer in which every neuron receives input from all neurons in the previous layer) to an exponential linear unit (ELU) activation function [24]. We choose the ELU over the commonly used rectified linear units (ReLU) activation function because it enables faster, more stable convergence during training, even

though it is less computationally efficient. Following the activation function, the last time step of the batch passes through the second dense layer, which outputs the predicted target variable for the given batch.

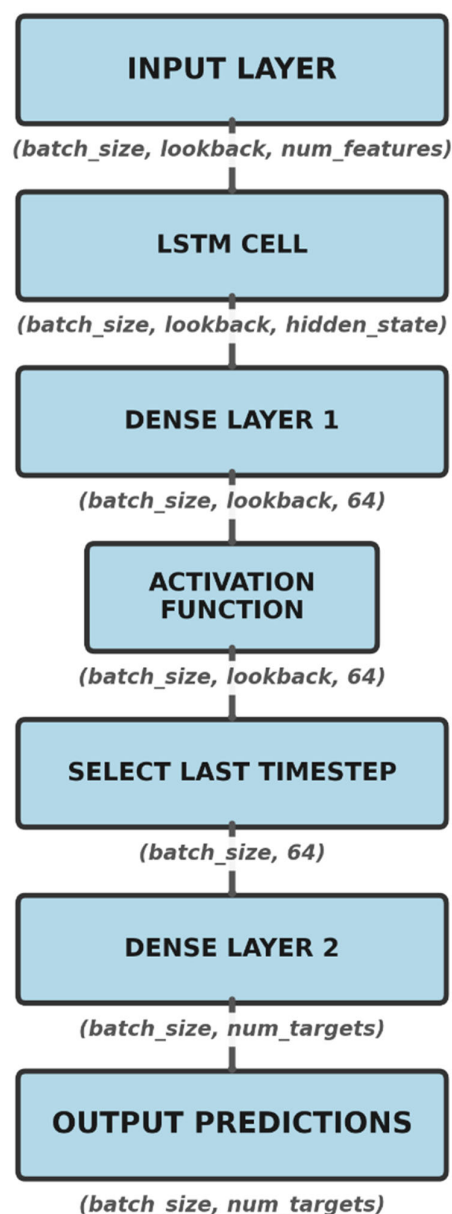


Figure 2. LSTM model architecture diagram. The output dimensions of each stage are presented in gray (see text for details).

Table 1 lists the default LSTM model hyperparameters we use. In each training iteration (out of the total number of epochs), the model processes a batch of samples from the training dataset following the process illustrated by the diagram in Figure 2 and ultimately updates its weights. We generally adjust the batch size and number of epochs while considering:

1. Exposing the model to the majority of training samples (given the effective training dataset size);
2. A reasonable total training duration;
3. Validation post-training that the model does not suffer from overfitting by examining the training curves (see Section 5).

Table 1. Default LSTM Model Hyperparameters.

Hyperparameter	Steady-State	Transition (Start-Up and Shut-Down)
Lookback [s]	90	90
LSTM cell hidden state size	128	128
Training batch size ¹	128	16
Number of training epochs ¹	125,000	85,000

¹ Adjustable depending on the effective training dataset size.

The hidden state size is set to 128 as a trade-off, allowing the model to effectively learn the various, relatively rich time series patterns while supporting fast prediction speed and manageable memory requirements. Finally, we implement a 90 s lookback to ensure models are exposed to at least two actual (not oversampled) measurements, assuming a minimum sampling resolution of 1 min.

4.3. Configuration of Individual LSTM Models

The default configuration of LSTM models we implement in the DT framework is shown in Figure 3. We train one set of models for steady-state operation and another for transition periods. With a single steady-state and transition model training taking up to 5 and 0.5 h to complete on a single, virtually split (shared) A100 graphics processing unit (GPU), respectively, training full sets of models requires less than 1.5 days in total (or a significantly shorter period when utilizing all GPU resources).

As input, each LSTM model receives the breaker circuit and/or governor status parameters, which are used downstream to inform the operational “online” mode which model (for steady-state or transition) to apply at a given instance or time step.

The LSTM model configurations are designed to associate the target variables only with physically related input variables (features). For example, on a given unit, the net head, which governs the penstock pressure, and wicket gate position are generally sufficient to robustly predict the generated power. (Note that in the case of Kaplan units, the runner blade pitch serves as an additional essential input variable.) For the LSTM model used to predict reactive power, we use the generated power as a proxy for load. Phase current is a good proxy of the load, in general. However, with future plans of the DT framework to actively incorporate the reactive power in the “offline” mode operations, and the unpredictable and highly load-dependent nature of the phase current, we prefer the generated power as a default alternative. The phase current can be used instead, if requested, but the generated power already exhibits satisfying results (not shown). For the terminal voltage model, we use a derived power factor, λ , defined here as

$$\lambda = \frac{P}{\sqrt{P^2 + Q^2}} \cdot \text{sign}(Q), \quad (1)$$

where P is the generated power in MW, and Q is the reactive power in MVAR.

Some of the models—namely, the gate position, penstock pressure, shaft speed, and terminal voltage models—have an auto-regressive component. That is, they receive their values in the preceding time steps as input (1 s lag), a method known as teacher forcing [25]. In the wicket gate position model, the lagged input is critical because the gate movement is constrained by a mechanical rate limit that is typically too slow to complete a position-change command within one or even several seconds. Therefore, during this movement, a model sampling at 1 s intervals must know the position in the preceding time step(s). In the case of the other three models partially implementing teacher forcing, the decision to use this method was a compromise stemming from:

1. Insufficient information received from the available sensors, which is required to capture and robustly predict the system dynamics. For example, a lack of information about how penstock water turbulence affects the pressure level or how dynamic grid load fluctuations influence shaft speed deviations.
2. Insufficient and/or inconsistent sampling rate needed to capture the system dynamics. For example, in the case of the terminal voltage and shaft speed.

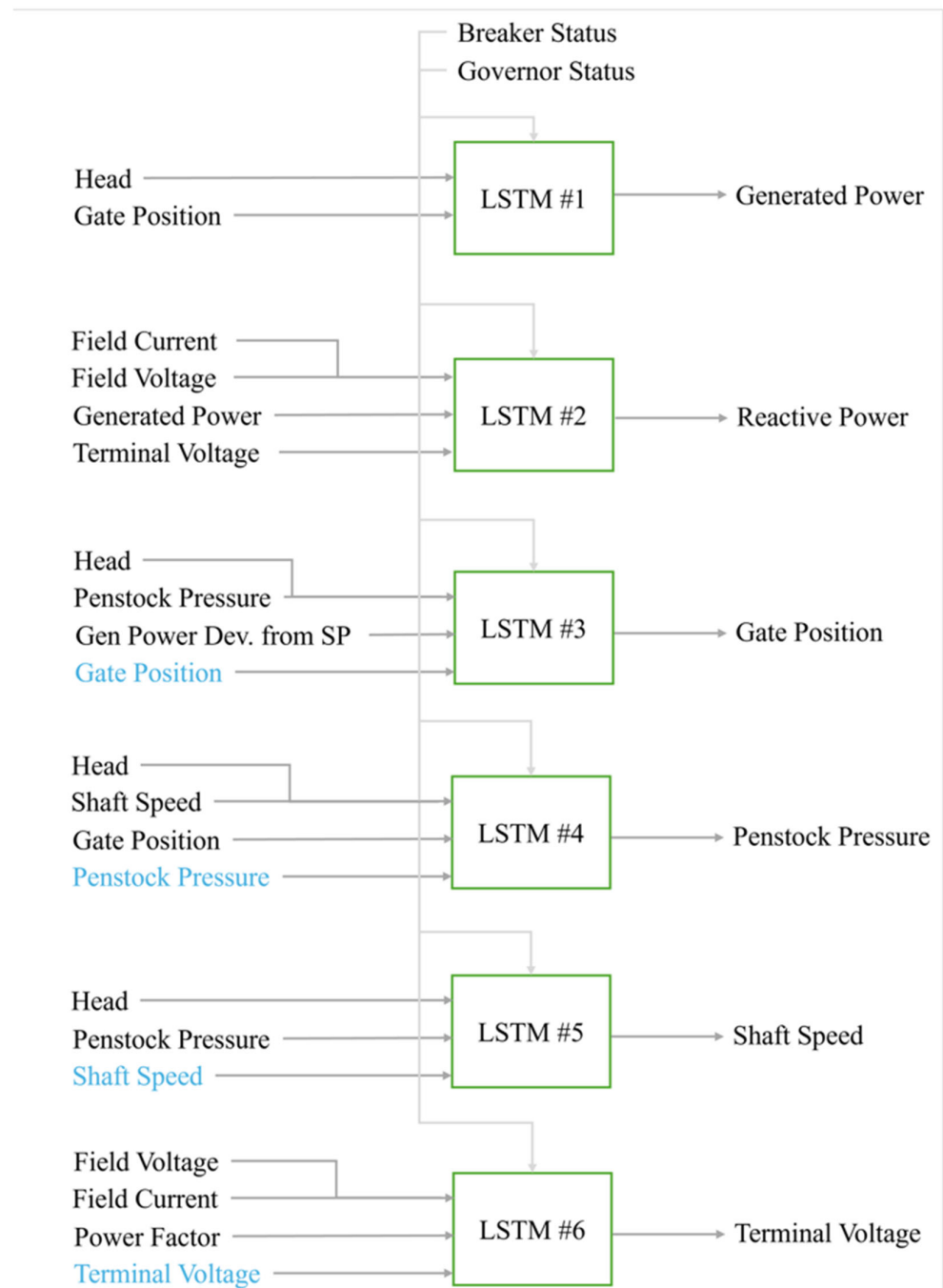


Figure 3. LSTM model input/output configurations. The blue font indicates variables that serve as both single-sample-lagged inputs and the target variable (teacher forcing). In the gate position model, “Gen Power Dev. Form SP” refers to the deviation of the generated power from the governor power set point.

The use of additional input variables for models operating the teacher-forcing method is intended to reduce, to some extent, dependence on lagged target variables, given the

tendency of such models to be unstable over consecutive predictions [26,27]. The power factor, used as a predictor in the terminal voltage model, serves as another proxy for load and, therefore, could have been a good predictor of reactive power. However, that introduces more complex teacher-forcing dependencies that we preferred to exclude from the model configuration. Finally, we note that sample differentials were tested as alternatives to direct teacher-forcing inputs because they could partially mitigate related limitations, but they did not produce a robust single-time-step prediction response that could have justified their implementation (not shown).

5. LSTM Model Evaluation

In this and the following sections, we use data from unit 12 at the Alder Dam facility, which is operated by Tacoma Public Utilities, on the Nisqually River, southeast of Tacoma, WA. We first use these data to demonstrate the LSTM model evaluation process.

Figure 4 exemplifies the learning curves we use to evaluate the LSTM models; in this case, the figure depicts the learning curves for the steady-state terminal voltage model. This and all other models for steady-state operation are trained on all steady-state samples collected in 2023. The testing/validation samples are based on data from January–June 2024. The depicted learning curves demonstrate a good fit, with the training and testing curves generally decreasing with epoch and becoming more stable towards the end of the training process. The two curves are generally aligned with each other in loss amplitude, and the sporadic “jumps” in loss amplitude become less frequent with epoch.

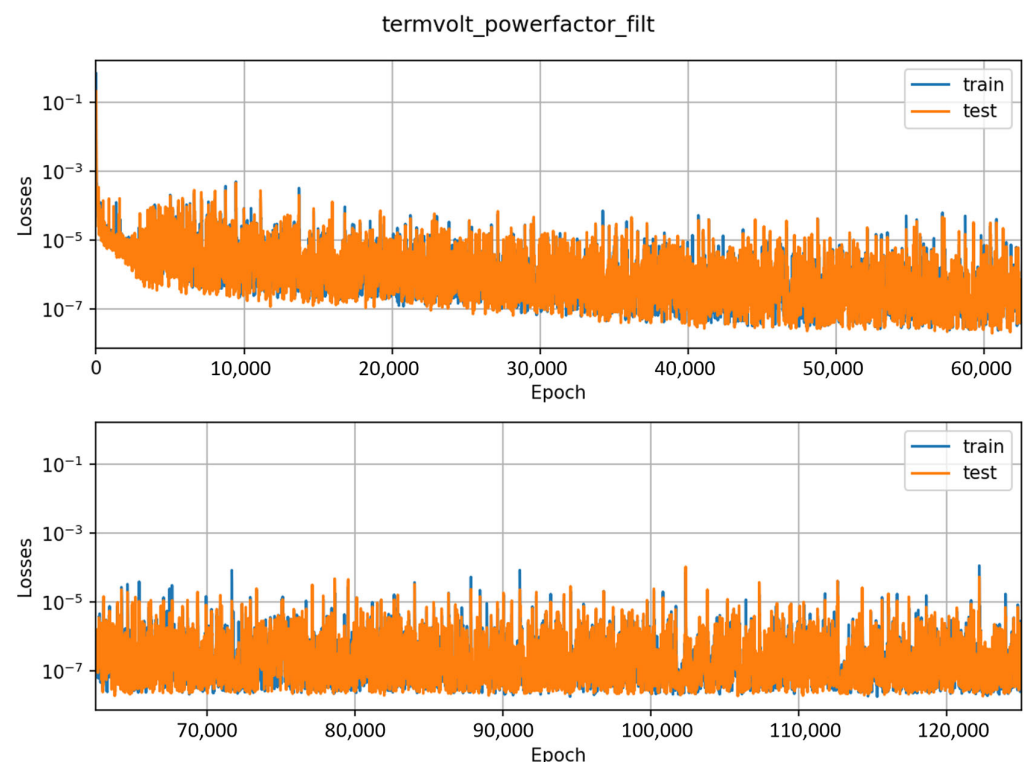


Figure 4. Learning curves for the unit 12 terminal voltage prediction LSTM model, illustrating the loss function (mean square error) as a function of training epoch (see legend). The bottom panel continues the top panel, covering all 125,000 training epochs.

The steady-state LSTM model predictions (red) for all six target variables are shown in Figure 5 for 26 March 2024, along with the observed values (black) and the relative prediction error (blue). The LSTM predictions of the generated power, reactive power, gate position, and penstock pressure exhibit good agreement with the observations, even

during sharp transitions (e.g., at gate positions). The models show very small relative errors, typically fractions of a percent. The terminal voltage predictions appear to closely follow the observed patterns, but with a persistent offset of ~ 0.012 kV. The shaft speed model also aligns with the observed patterns, but with persistent offset and scaling factor biases. Scaling those biases yields good agreement with the observed patterns (not shown).

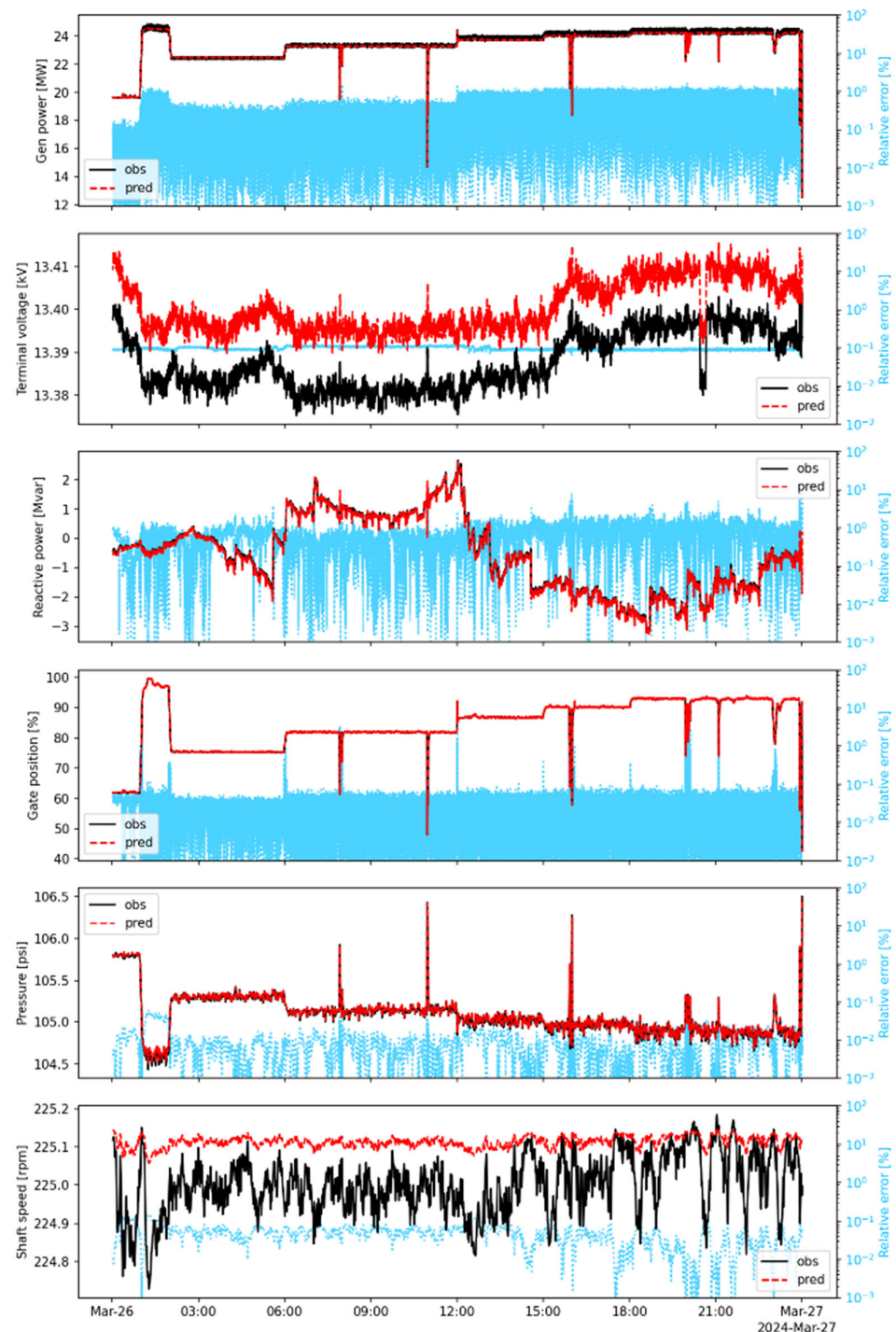


Figure 5. LSTM model predictions (all for steady-state operation status) for unit 12 on 26 March 2024, showing (from the top down) generated power, terminal voltage, reactive power, gate position, penstock pressure, and shaft speed. True measurements are illustrated in black, predictions in red, and instantaneous relative prediction errors in blue.

A more extensive statistical analysis of model predictions is shown in Figure 6. The depicted mean square error (MSE), mean absolute error (MAE), and root-mean-square error (RMSE) are calculated with normalized samples; hence, error units are the same for all target variables.

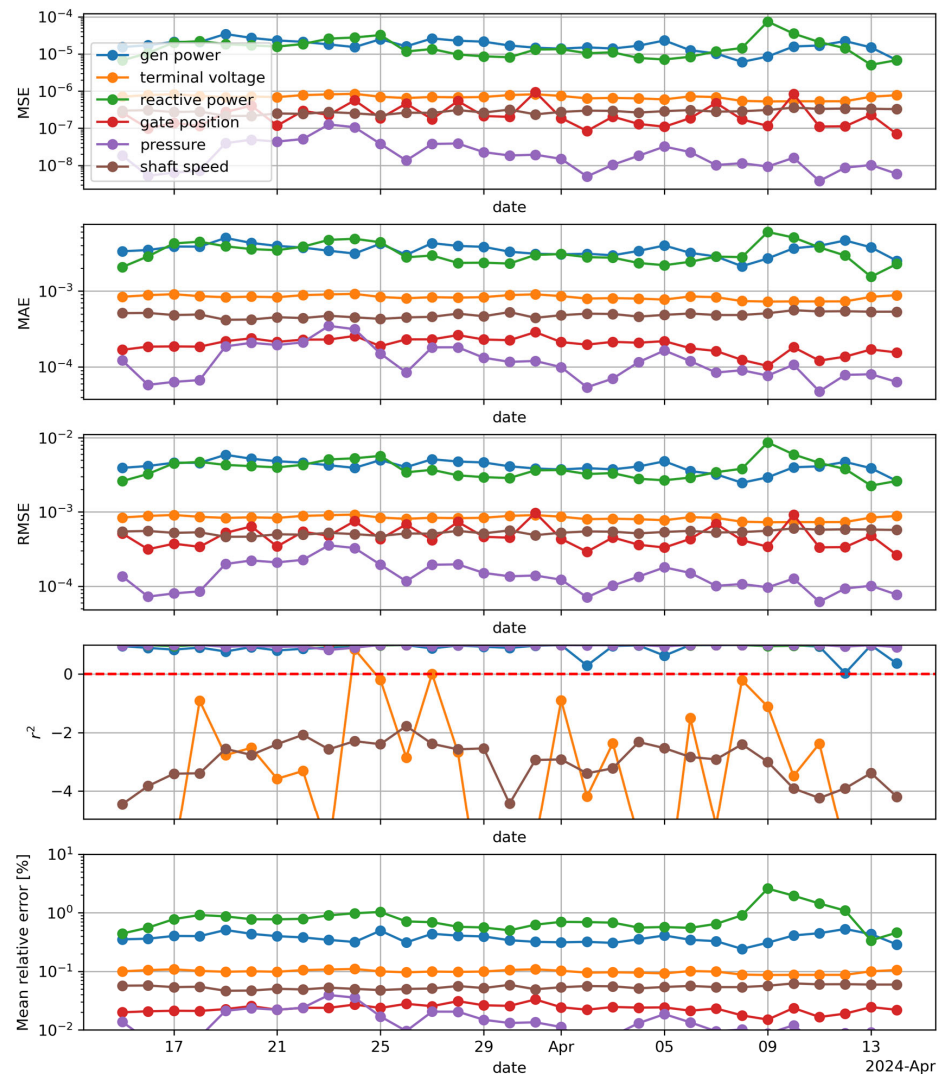


Figure 6. Statistical analysis of each model's steady-state predictions for unit 12 (see legend), covering a 1-month period between 15 March 2024, and 14 April 2024, during which the examined unit was operated continuously. From the top down: normalized mean square error (MSE), normalized mean absolute error (MAE), root-mean-square error (RMSE), coefficient of determination (r^2), and the mean relative error. Those metrics are calculated separately using all relevant samples on a given day. The dashed red line on the r^2 panel delineates a value of 0, with samples below it indicating a lack of predictive capability relative to a fixed-sample-average-value model.

Consistent with the single-day example shown in Figure 5, all models demonstrate small errors, with mean relative errors smaller than 1% in the case of the generated power and reactive power models, except for the 9–12 April period, for which the reactive power model shows relatively larger errors of up to 3%. All other models show mean relative errors of ~0.1% or lower. As expected, the daily patterns of the MSE, MAE, and RMSE are similar and largely consistent with those of the mean relative error.

The coefficient of determination (r^2) for the generated power, reactive power, gate position, and penstock pressure models is close to 1, indicating strong predictive capability. However, on certain dates (2, 5, 12, and 14 April), the generated power model shows a smaller

r^2 , although still greater than 0, suggesting some predictive capability. These lower r^2 instances are attributable to a persistent positive prediction bias of ~ 0.1 MW. (Note that r^2 is used in its general form here—not the specific case of a linear model fit—hence, it can become negative, and it decreases as a result of biases, even if general prediction patterns match). For the equivalent argument, and consistent with Figure 5, the application of the terminal voltage and shaft speed models results in negative r^2 values, indicating a lack of predictive capability for absolute values, a likely outcome of their auto-regressive component. We note that scaling of these variables on the basis of the examined period statistics rectifies r^2 to high positive values, reflecting the generally well-captured variability patterns.

Figure 7 depicts the application of the same statistical analysis to transition operation status periods between January and May 2025. Trained on all transition periods between 2020 and 2024, the LSTM models primarily exhibit reasonable predictive skill, with small errors (though larger than the equivalent steady-state models) and mostly positive r^2 values for the penstock pressure and shaft speed. Manual evaluation of these models (not shown) suggests that they are able to capture well the speed slowdown and ramp-up during shut-down and start-up operation, respectively, as well as the associated pressure changes.

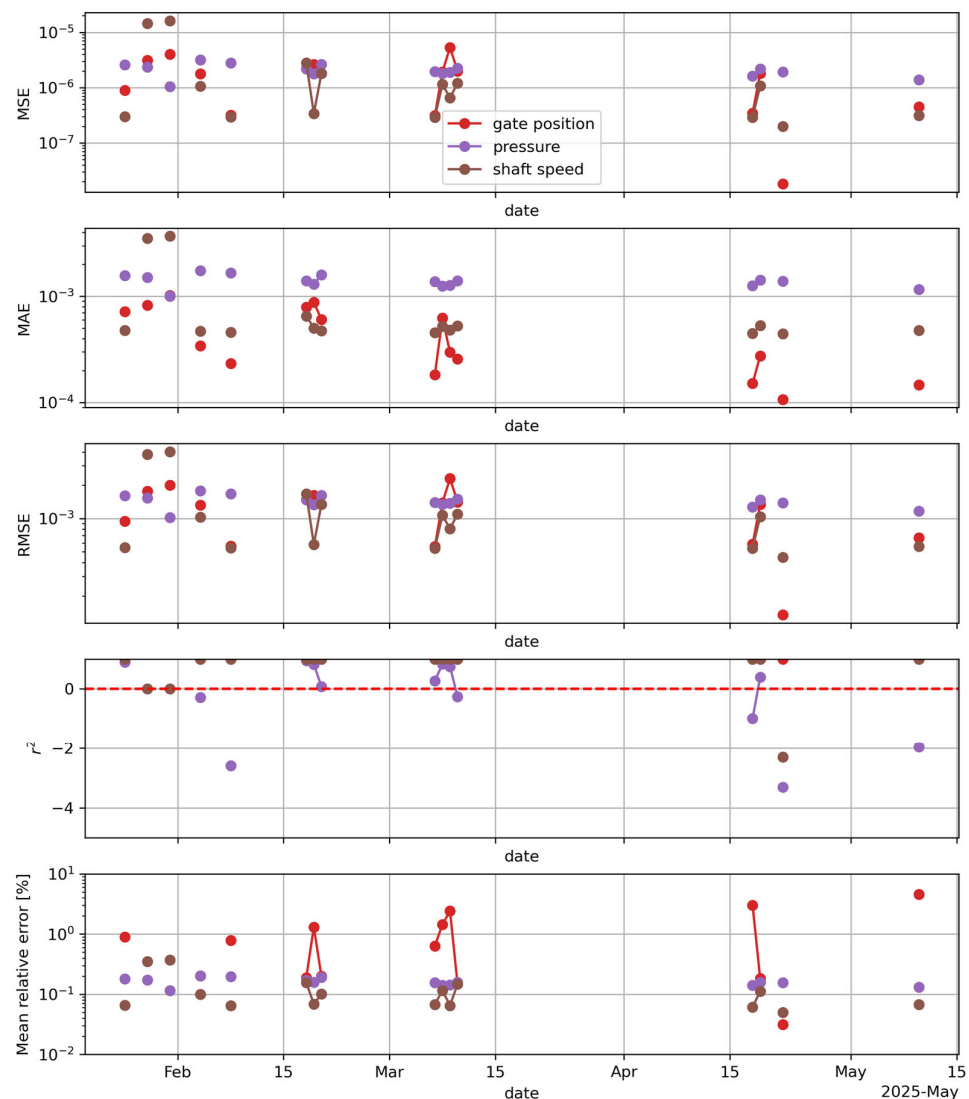


Figure 7. As in Figure 6, but for transition operation status periods between January and May 2025. Lines connecting the sample points indicate consecutive days on which unit transition (shut-down or start-up) occurred.

The model evaluation results indicate that the LSTM models trained as part of the DT framework can provide accurate predictions with small errors. Assuming these properties apply to the vast majority of operating conditions (not all plausible conditions were trained or tested on), we suggest that these models provide a strong basis for system monitoring via continuous, real-time, single-sample predictions (“online” mode). However, small as the errors may be, error compounding over a large number of recursive iterations (e.g., when operating the “offline” mode), combined with the complex non-linear dynamics of certain aspects of hydropower units (e.g., the excitation system) could result in substantial prediction errors, especially when applying those models trained using teacher-forcing methods. Therefore, “offline” models should be limited in their offered applicability and constrained to mitigate such effects. In the following section, we portray how those capabilities and limitations come into play in the operational DT data portal and demonstrate how some of the trained models, especially those not trained using the teacher-forcing method (e.g., the generated power model), can also be used to help site operators with system-characterization tasks.

6. DT Framework Applications

6.1. Unit Characterization

One of the tasks the trained LSTM models can be used for is characterizing the generated power as a function of its predictors—namely, the net head and the wicket gate position (in the case of Francis-type units). Figure 8 shows the predicted generated power mapped to the net head and gate position parameter space. Although most of the depicted parameter space can be largely characterized by means of interpolation of actual observations, the LSTM model is able to extend the applicable parameter space to out-of-sample predictions.

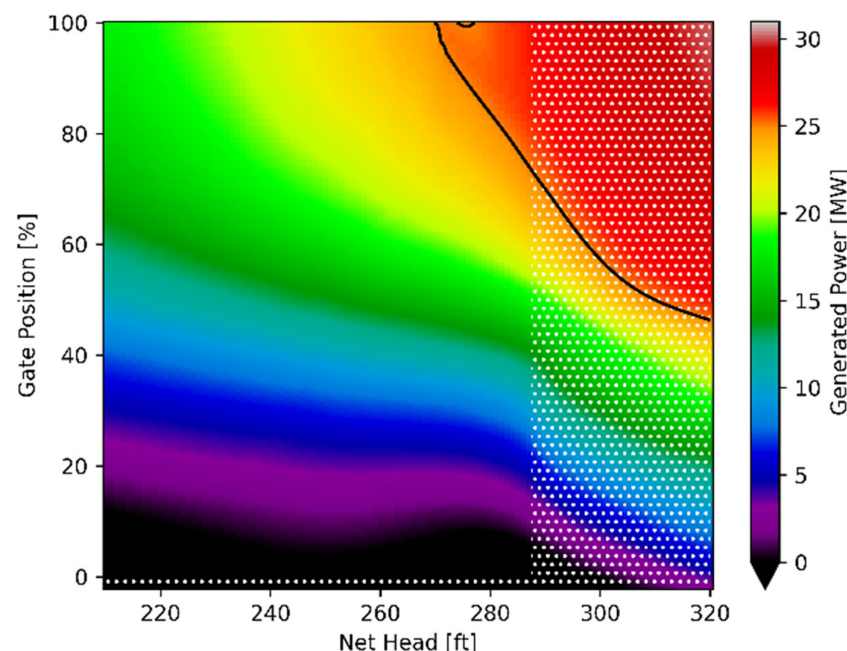


Figure 8. Predicted generated power for unit 12 as a function of wicket gate position and net head based on the trained LSTM model. The hatched areas represent out-of-sample ranges of the training dataset parameter space determined separately for each axis. The black contour denotes 25 MW, the unit 12 production limit.

This analysis output can be used to demonstrate behavior predicted on the basis of proposed changes in governor control inputs and in tasks such as optimizing operational

fits for application in the governor code. Alternatively, these results can be used to examine system degradation over time through analysis of an equivalent power generation dataset collected at a later point in time, assuming that the relevant LSTM model used as a reference was trained on data collected under nominal operating conditions. Similarly, these results could also be used to quantify the value of maintenance practices with respect to improvements to this type of chart. With our DT framework's models, various parameter-space exploration tasks for different applications can be performed, beyond the scope of this study, depending on site operator needs.

6.2. Routine (“Online”) Mode

The “online” mode, in which the LSTM models are applied continuously to data ingested at near-real time, is the fully data-driven mode of the DT framework. In this mode, the LSTM models provide predictions that are independent of those of previous and subsequent time steps (other than the utilized lookback period used by the LSTM algorithm itself). This operation mode serves as the basis for alarm options (future development), in which temporal regions are highlighted in case of observations that continuously deviate from the expected values, determined from the LSTM predictions.

Figure 9 shows a screenshot of the “online” mode viewer on the DT data portal, depicting the measured unit 12 data together with the LSTM model predictions. Users can smoothly zoom in and out, toggle different parameters, and slide through the temporal ranges.



Figure 9. Screenshot from the “online” mode viewer on the DT data portal showing the LSTM models’ outputs for the U12 unit. The depicted period covers 10 September 2023, through 6 October 2023.

6.3. Simulation (“Offline”) Mode

The DT’s “offline” model enables site operators to run simulations of unit power response to its governing system and supports simulation comparison of different proposed control-loop modifications. In this mode, we effectively use a hybrid approach that has

some structural similarity to the approach outlined in [11]. In this hybrid approach, the generated power prediction is performed using the unit's LSTM power model, but a Python code designed to replicate the control response of the field governor is utilized to interact with the generator response. For example, in the case of Alder Dam's unit 12 demonstrated here, we use code emulating a HYG3 governor [28] modified with feedforward control as observed in the governor for the field unit. We also emulate a simplified version of the ESST1A excitation system operated by the unit [28]; however, as mentioned above, we overlook its output at this development stage, mainly because of data acquisition limitations and the general complexity of excitation system codes (see discussion at the end of Section 5).

Using the calculated Python response from our governor emulator, we can feed the model parameters, such as closed-loop control gains and response deadbands, and therefore tune the unit's governor without physically operating it. Figure 10 shows a screenshot from the DT “offline” mode dashboard, where the user can set up and configure a hydropower unit simulation, control parameters such as gains, and choose “replay” mode or a customized power demand chart.

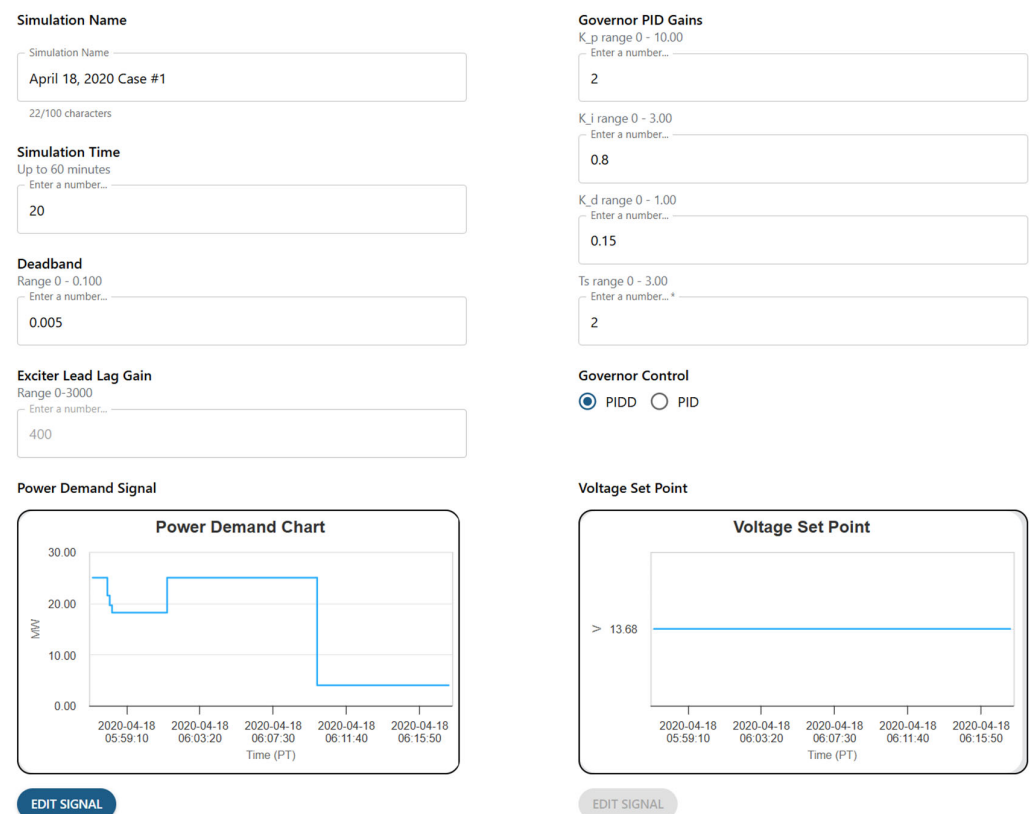


Figure 10. Screenshot from the “offline” mode simulation setup on the digital twin data portal. In this case, a reproduction of a real U12 case from 18 April 2020.

Given that the governor response requires the first- and second-order time derivative of the shaft speed information to calculate integrated errors, and that the shaft speed model can capture its temporal evolution, albeit to a more limited extent and with an offset, we implement the LSTM models in the “offline” mode to facilitate smooth operation. The shaft speed model output is tuned to correct for its prevailing bias (0.12 rpm in the case of unit 12) and is constrained to a typical range in the real data to ameliorate cases in which the shaft speed predictions deviate or drift from this expected offset. The constrained range is set to 224.9–225.1 rpm for unit 12.

Figure 11 shows a screenshot of the simulation output corresponding to the configuration depicted in Figure 10. Similar to the “online” mode viewer, the “offline” mode viewer enables the user to toggle between variables and examine the variability of the illustrated variables by hovering over the curves. This case demonstrates the robustness of the recursive LSTM-generated power predictions, with relative errors of less than 4% in magnitude. Thus, this DT operation mode facilitates counterfactual exercises to determine the unit’s response to various scenarios, such as different net head levels, power demands, and other configurable parameters.

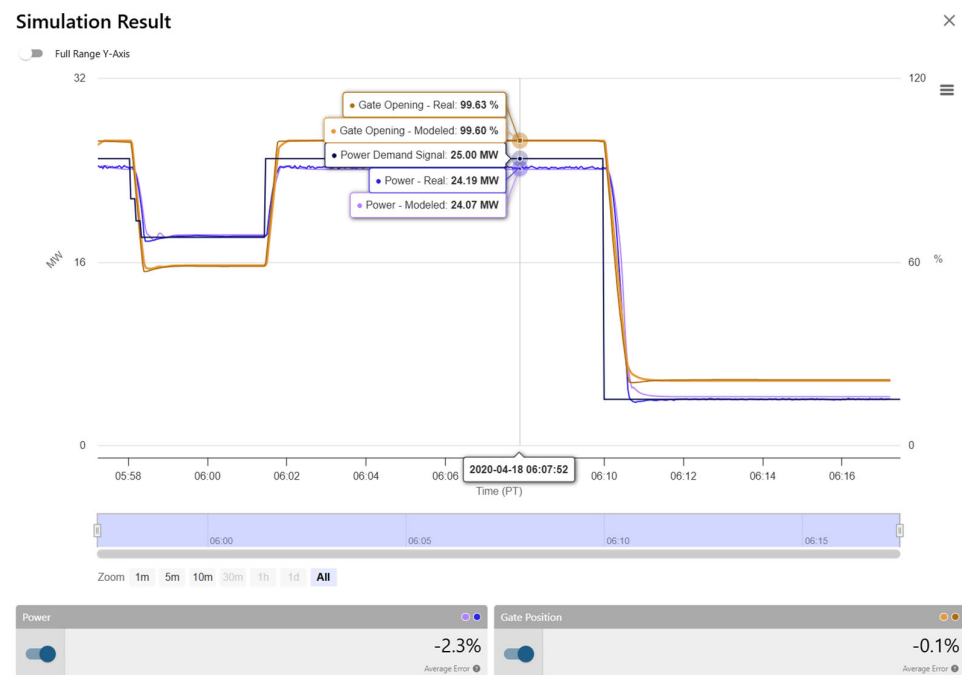


Figure 11. Screenshot from the “offline” mode simulation output presented on the digital twin data portal for the unit 12 case configured and shown in Figure 10.

To further enhance the user experience, the DT data portal includes a simulation comparison feature that enables users to evaluate the results of two offline simulations side-by-side using interactive line charts. Users can select any pair of simulations configured with the same simulation time and visualize the values of key variables, such as generated power and gate position, plotted within a single chart. Presenting the data from both simulations in a single view streamlines the identification of operational responses, trends, patterns, and discrepancies between scenarios.

Figure 12 shows a screenshot of the simulation comparison dashboard. The comparison feature includes interactive controls, such as a dynamic legend that allows users to show or hide lines corresponding to specific variables, whereas the charts offer zooming and panning functions, facilitating closer examination of specific time intervals. Similar to the single simulation output viewer, users can also export the results in multiple formats (e.g., PNG images, CSV tables, or PDF documents) to support activities external to the data portal, such as meetings or follow-up analyses.

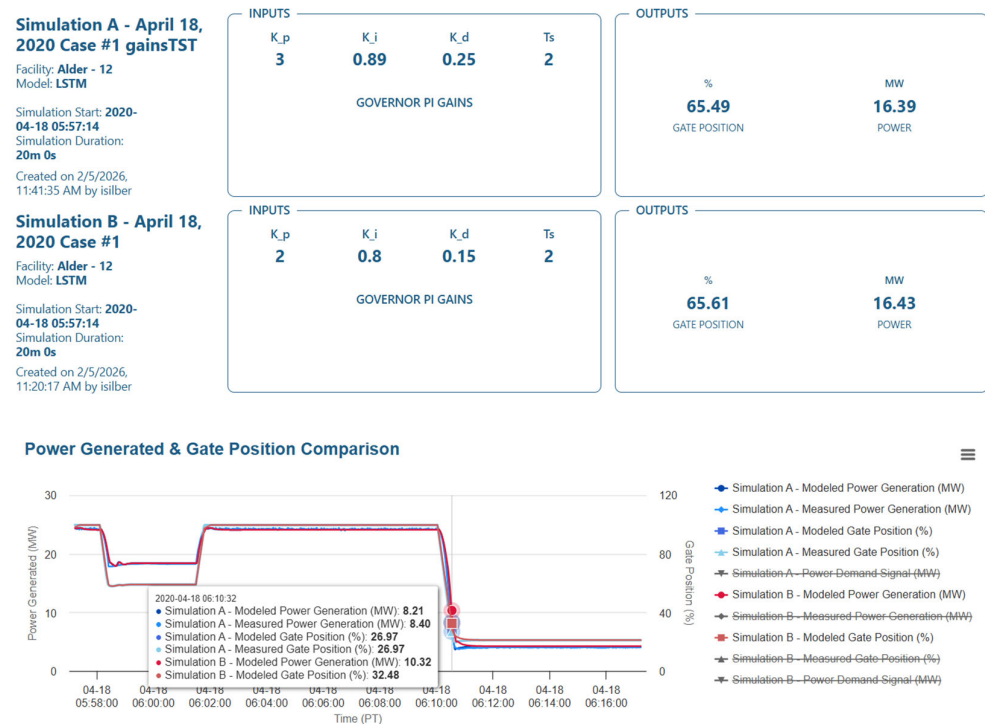


Figure 12. Screenshot from the “offline” mode simulation comparison dashboard. In this case, two “offline” simulations are compared for the same unit 12 case depicted in Figures 10 and 11: The first simulation is the same one shown in Figure 11, whereas the second simulation is configured using modified closed-loop control gains.

7. Conclusions and Outlook

Digital twin (DT) frameworks leverage data-driven models to optimize energy production, detect anomalies and component degradation, and trigger early-warning alarms. DT frameworks not only improve operational efficiency and asset longevity but also strengthen energy dominance in this critical infrastructure by minimizing costly downtimes. We demonstrated an operational DT framework integrated into an easy-to-use data portal designed with site operators in mind. The LSTM approach cuts overhead relative to other more costly DT approaches, and facilitates framework scalability. A significant number of the DT features and components demonstrated here are already automated and integrated in our user-friendly data portal. Other features, such as data characterization and the generation of training datasets, are streamlined but not fully automated, given the need for domain experts and expert evaluation.

Our DT framework is already scalable and generally adaptive. However, we envision new features and expansions of this framework, some of which are already in development and testing. Those expansions include, among others, implementation of automatic alarms customized to customer needs; integration of unit-level models into full-facility (common bus) meta-models; merging automated and periodic short-term “offline” mode simulations into the “online” mode to facilitate more robust anomaly detection capabilities including failure prediction; refining the “offline” mode and potentially expanding its capabilities to capture a greater fraction of the excitation system dynamics (depending on customer needs) by controlling the auto-regressive (teacher forcing) models via a meta model (e.g., see Lamb [26]); enabling the prescription and optimization of maintenance outages; and automation of components using agentic AI. We invite interested parties to contact us, examine the DT dashboard (at <https://hydro.digitaltwin.labworks.org/>; accessed on 29 July 2026), collaborate, and/or provide feedback.

Author Contributions: Conceptualization, I.S., S.W. and C.S.; methodology, I.S. and S.W.; software, I.S., S.W., M.R.K., S.B. (Shuhao Bai), S.B. (Sherman Beus) and M.C.M.; validation, I.S. and M.R.K.; formal analysis, I.S.; investigation, I.S. and S.W.; data curation, I.S., M.R.K. and S.B. (Sherman Beus); writing—original draft preparation, I.S.; writing—review and editing, all authors; visualization, I.S.; funding acquisition, C.S.; project administration, C.S. All authors have read and agreed to the published version of the manuscript.

Funding: This project was funded by the Hydropower and Hydrokinetics Office of the U.S. Department of Energy under the Development of a Digital Twin for Hydropower Systems project. Pacific Northwest National Laboratory is operated for the U.S. Department of Energy by Battelle under contract DE-AC05-76RL01830.

Data Availability Statement: The datasets presented in this article are not readily available because the data are proprietary in nature and are part of a Non-Disclosure Agreement with the collaborating utility companies; we do not have permission to share the results with a wider public audience. Requests to access the datasets should be directed to the DT framework’s data portal for the utility operators at <https://hydro.digitaltwin.labworks.org/> (data access is limited to members who have signed the agreement).

Acknowledgments: We acknowledge Kyle DeSomber for the valuable feedback on the manuscript.

Conflicts of Interest: Author Greg Kenyon was employed by the company Tacoma Public Utilities. The remaining authors are from Pacific Northwest National Laboratory, which is operated for the U.S. Department of Energy by Battelle. The authors declare that this study received funding from the U.S. Department of Energy. The funder was not involved in the study design, collection, analysis, interpretation of data, the writing of this article or the decision to submit it for publication.

Abbreviations

The following abbreviations are used in this manuscript:

DT	Digital twin
ELU	Exponential linear unit
GPU	Graphics Processing Unit
LSTM	Long short term memory
MAE	Mean absolute error
MSE	Mean square error
P	Generated power
PNNL	Pacific Northwest National Laboratory
Q	Reactive power
ReLU	Rectified linear unit
RMSE	Root mean square error
SG	Savitzky–Golay
λ	Power factor

References

1. U.S. Energy Information Administration *Monthly Energy Review*; U.S. Department of Energy: Washington, DC, USA, 2023.
2. Uria Martinez, R.; Johnson, M. *U.S. Hydropower Market Report 2023*; p. ORNL/SPR–2023/3076, 2006921; Oak Ridge National Laboratory (ORNL): Oak Ridge, TN, USA, 2023.
3. Turner, S.W.D.; Ghimire, G.R.; Hansen, C.; Singh, D.; Kao, S.-C. Hydropower Capacity Factors Trending down in the United States. *Nat. Commun.* **2024**, *15*, 5445. [\[CrossRef\]](#)
4. Zhao, Z.; Li, D.; She, J.; Zhang, H.; Zhou, Y.; Zhao, L. Construction and Application of Digital Twin Model of Hydropower Plant Based on Data-Driven. In *Proceedings of the 2021 3rd International Workshop on Artificial Intelligence and Education (WAIE)*; IEEE: Xi’an, China, 2021; pp. 60–64.
5. Kim, M.; Ghobadi, F.; Tayerani Charmchi, A.S.; Lee, M.; Lee, J. Digital Twins for Clean Energy Systems: A State-of-the-Art Review of Applications, Integrated Technologies, and Key Challenges. *Sustainability* **2025**, *18*, 43. [\[CrossRef\]](#)

6. Chen, W.; Su, Y.; Mathew, A.; Brilakis, I. Optimising Hydropower Operations: A Tailored Digital Twin Framework for Hydropower Plant. In *Proceedings of the Sixth International Conference on Civil and Building Engineering Informatics (ICCBEI 2025)*; EasyChair: Stockport, UK, 2025; pp. 767–777.
7. Tubeuf, C.; Birkelbach, F.; Maly, A.; Hofmann, R. Increasing the Flexibility of Hydropower with Reinforcement Learning on a Digital Twin Platform. *Energies* **2023**, *16*, 1796. [\[CrossRef\]](#)
8. Tavares, M.; Pérez-Sánchez, M.; Carravetta, A.; Coronado-Hernández, O.E.; López-Jiménez, P.A.; Ramos, H.M. Smart Feasibility Optimization of Hybrid Renewable Water Supply Systems by Digital Twin Technologies: A Multicriteria Approach Applied to Isolated Cities. *Sustain. Cities Soc.* **2024**, *115*, 105834. [\[CrossRef\]](#)
9. Cai, Z.; Wang, Y.; Zhang, D.; Wen, L.; Liu, H.; Xiong, Z.; Wajid, K.; Feng, R. Digital Twin Modeling for Hydropower System Based on Radio Frequency Identification Data Collection. *Electronics* **2024**, *13*, 2576. [\[CrossRef\]](#)
10. Guo, J.; Zhang, Y.; Wang, W. Predicting Hydro Turbine Failures through Digital Twin Simulations of Rare Real-World Data. *J. Comput. Methods Sci. Eng.* **2025**, 14727978241304260. [\[CrossRef\]](#)
11. Wang, H.; Ou, S.; Dahlhaug, O.G.; Storli, P.-T.; Skjelbred, H.I.; Vilberg, I. Adaptively Learned Modeling for a Digital Twin of Hydropower Turbines with Application to a Pilot Testing System. *Mathematics* **2023**, *11*, 4012. [\[CrossRef\]](#)
12. Wang, H.; Yin, Z.; Jiang, Z.-P. Real-Time Hybrid Modeling of Francis Hydroturbine Dynamics via a Neural Controlled Differential Equation Approach. *IEEE Access* **2023**, *11*, 139133–139146. [\[CrossRef\]](#)
13. Machalski, A.; Szulc, P.; Błoński, D.; Nycz, A.; Nemš, M.; Skrzypacz, J.; Janik, P.; Satława, Z. The Concept of a Digital Twin for the Wały Śląskie Hydroelectric Power Plant: A Case Study in Poland. *Energies* **2025**, *18*, 2021. [\[CrossRef\]](#)
14. Zeng, Y.; Hussein, Z.A.; Chyad, M.H.; Farhadi, A.; Yu, J.; Rahbarimagham, H. Integrating Type-2 Fuzzy Logic Controllers with Digital Twin and Neural Networks for Advanced Hydropower System Management. *Sci. Rep.* **2025**, *15*, 5140. [\[CrossRef\]](#)
15. Tan, J.; Radhi, R.M.; Shirini, K.; Gharehveran, S.S.; Parisooz, Z.; Khosravi, M.; Azarinfar, H. Innovative Framework for Fault Detection and System Resilience in Hydropower Operations Using Digital Twins and Deep Learning. *Sci. Rep.* **2025**, *15*, 15669. [\[CrossRef\]](#)
16. Wang, H.; Ou, S. Structured Neural Network Modeling for Developing Digital Twins Models of Hydropower Generation Units. In *Proceedings of the 2024 IEEE 19th Conference on Industrial Electronics and Applications (ICIEA)*; IEEE: Kristiansand, Norway, 2024; pp. 1–6.
17. Ersan, M.; Irmak, E. Development and Integration of a Digital Twin Model for a Real Hydroelectric Power Plant. *Sensors* **2024**, *24*, 4174. [\[CrossRef\]](#)
18. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [\[CrossRef\]](#)
19. Savitzky, A.; Golay, M.J.E. Smoothing and Differentiation of Data by Simplified Least Squares Procedures. *Anal. Chem.* **1964**, *36*, 1627–1639. [\[CrossRef\]](#)
20. Heek, J.; Levskaya, A.; Oliver, A.; Ritter, M.; Rondepierre, B.; Steiner, A.; van Zee, M. *Flax: A Neural Network Library and Ecosystem for JAX 2024*; GitHub: Mountain View, CA, USA, 2024.
21. Frostig, R.; Johnson, M.J.; Leary, C. Compiling Machine Learning Programs via High-Level Tracing. In *Proceedings of the SysML Conference 2018*; MLSys Organization: Stanford, CA, USA, 2019.
22. Xiao, J.; Deng, T.; Bi, S. Comparative Analysis of LSTM, GRU, and Transformer Models for Stock Price Prediction. In *Proceedings of the International Conference on Digital Economy, Blockchain and Artificial Intelligence*; ACM: Guangzhou, China, 2024; pp. 103–108.
23. Zeng, A.; Chen, M.; Zhang, L.; Xu, Q. Are Transformers Effective for Time Series Forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI Press: Washington, DC, USA, 2023; Volume 37, pp. 11121–11128. [\[CrossRef\]](#)
24. Clevert, D.-A.; Unterthiner, T.; Hochreiter, S. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). In *Proceedings of the 4th International Conference on Learning Representations (ICLR 2016)*; OpenReview.net: Amherst, MA, USA, 2016.
25. Williams, R.J.; Zipser, D. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Comput.* **1989**, *1*, 270–280. [\[CrossRef\]](#)
26. Lamb, A.; Goyal, A.; Zhang, Y.; Zhang, S.; Courville, A.; Bengio, Y. Professor Forcing: A New Algorithm for Training Recurrent Networks. In *Proceedings of the Advances in Neural Information Processing Systems 29*; Curran Associates, Inc.: Red Hook, NY, USA, 2016.
27. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
28. Koritarov, V.; Guzowski, L.; Feltes, J.; Kazachkov, Y.; Lam, B.; Grande-Moran, C.; Thomann, G.; Eng, L.; Trouille, B.; Donalek, P. *Review of Existing Hydroelectric Turbine-Governor Simulation Models*; p. ANL/DIS--13/05, 1098022; Argonne National Laboratory (ANL): Lemont, IL, USA, 2013.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.