Reactor dynamics surrogate models for digital twin - A case for ARIMAX models and a case against LSTMs

Jasmin Lim, Karthik Duraisamy

Department of Aerospace Engineering, University of Michigan
1320 Beal Ave, Ann Arbor, MI 48109, USA
{jaslim, kdur}@umich.edu

Abstract

Nuclear energy has high operational risks that a digital twin can aid in overall management, providing safety analysis and increase operational reliability. At the core of the reactor digital twin is an accurate surrogate model of the plant dynamics. The reactor dynamics surrogate model should be fast and accurate to provide data for subsequent models, predictive maintenance, and control optimization. We compare the Long Short-Term Memory (LSTM) network and Autoregressive Integrated Moving Average with Extra Input (ARIMAX) model for the nuclear reactor plant dynamics. The LSTM uses gates to filter and retain information in an existing cell state, making it attractive for sequential data with long term dependencies. However, the complex structure of the LSTM requires training thousands of parameters with a large dataset. Therefore, we consider at the comparatively simpler ARIMAX model, which determines future state prediction as a function of past states and residual errors with integrators. For a simple single input-single output case, ARIMAX shows far better accuracy in comparison to LSTMs, with the latter not able to capture the physical structure of the dynamical response. ARIMAX takes less time for parameter training and unseen data prediction, making it more suitable for digital twin integration.

Introduction

Nuclear reactors are highly complex systems that demand around the clock monitoring and management. The development of a digital twin is in efforts of making the operational tasks of nuclear energy; 1) more affordable by its potential to reduce maintenance costs through service and degradation predictions; and 2) safe, through real-time reactor monitoring and analysis.

The Molten Salt Reactor (MSR) is a Generation IV nuclear reactor (Locatelli et al. 2013) that uses a salt coolant and graphite moderator. These salt coolants have high thermal efficiency and high boiling points, allowing the reactors to operate at high temperatures while maintaining low pressure. A pebble-bed fluoride-salt-cooled high temperature reactor (FHR) is a type of MSR that uses the pebble form of TRISO fuel combined with a fluoride salt coolant (Hu et al. 2020). The fuel pebbles have excellent chemical stability and a high heat capacity for heat transfer at high temperatures (Zhao et al. 2022). This makes the FHR safer to operate at higher temperatures and improves the retaining of fission product.

In order for a digital twin to provide a reliable and accurate simulation of the physical asset, a surrogate model for the whole-plant reactor dynamics needs to be in place. Decisions about operational control, management, and maintenance will depend on predictions from the surrogate model about the system state response to inputs. The system state includes measured quantities (e.g. temperature, pressure, mass flow rate), derived process variables (e.g. heat extraction rate by the heat exchanger), and unobserved variables (e.g. reactivity feedbacks). The surrogate model needs to predict these values for real-time configuration analysis.

The performance of a neural network and autoregressive model are compared as surrogates for the reactor dynamics. We implement a Long Short-Term Memory Neural Network (Hochreiter and Schmidhuber 1997) and a Autoregressive Integrated Moving Average with Extra Input (ARIMAX) model to see which provides accuracy while minimizing computational costs.

Standard feed-forward Neural Networks are a common deep neural network framework where an input traverses a series of hidden layers to formulate an output. However, in this structure, the inputs and outputs are assumed to be independent. Recurrent Neural Networks (RNNs) have internal cycle structures where network inputs are functions of network outputs, exhibiting an input-response like behavior found in dynamical time-series. LSTM networks are a subclass of RNNs that can be used to predict sequential data with long-term dependencies by using a continuously existing cell state. Information is added or removed from the cell state through a filtering gate structure. There are three gates in the LSTM cell: the forget gate; the input gate; and the output gate. The forget gate determines if information is removed from the cell state. Next, new information is filtered through the input gate and it is decided if it will be stored in the cell state. Finally, the output gate filters the updated cell state and pushes it to the next sequential step. Each LSTM cell propagates the important information forward to be influential on future states.

Similarly to LSTMs, the ARIMAX model assumes the dependency of past states on future states, thus it is formulated such that the state at future time steps is a linear combination of the state at past time steps. The standard Autoregressive Moving Average (ARMA) model describes a stationary stochastic process in terms of two polynomials: the autoregressive using lagged values; and the moving average that leverages the current and historical error. The ARMA model is extended to the ARMAX where X refers to exogenous variables, and the model can account for current and past external inputs. However, in practical real-life processes, the external inputs to the system will change the local mean and variance of the states. Thus, for non-stationary processes, we further extend the ARMAX model to the ARIMAX model. In this version, integration is performed on the data to make it behave like a stationary process so that the ARMAX formulation can be used.

We compare the performance of the LSTM and the ARIMAX on their ability to provide accurate predictions of key reactor state values due to input changes. We also measure their computational costs with time, and the model complexity with the number of trainable parameters.

Methods

System Analysis Module

The System Analysis Module (Hu et al. 2021) is a system analysis tool for the safety analysis of advanced non-light-water reactors (non-LWR), this includes FHRs. The System Analysis Module (SAM) is currently being developed by Argonne National Laboratory to provide a fast, intermediate-fidelity simulation, with whole-plant transient capabilities for quick design analysis of advanced reactor concepts.

SAM is used to simulate the reactor dynamics of an FHR plant configuration, shown in Figure 1. The model is comprised of a Point Kinetics (PKE) reactor power core, a primary loop pump and counter-current flow heat exchanger between the primary FLiBe salt loop and the secondary FLiNaK salt loop. The secondary side of the heat exchanger has boundary conditions on inlet mass flow rate $v_{s,BC} = 0.39$ m/s, inlet temperature $T_{s,BC} = 765$ K, and the outlet pressure $P_{s,BC} = 1 \times 10^5$ Pa.

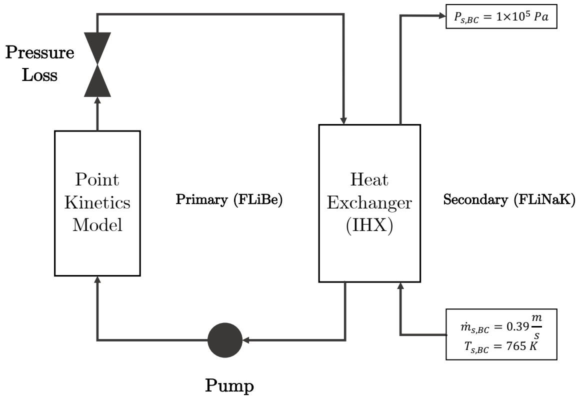

Figure 1: PKE reactor dynamics of SAM simulation for data generation.

Data Generation

To train the surrogate model, SAM generates data of the plant dynamics with input changes to the primary pump head. The initial pressure $H_{P,init} = 56,321.83$ Pa is modified to either increase or decrease by $M\%$ in a ramp time of R seconds. Both M and R are selected at random from a uniform distributions; the modification percentage is $M \sim \mathcal{U}(0, 20)$ and the ramp time is $R \sim \mathcal{U}(300, 1200)$.

The total simulation time is 4000 seconds, with the time step $\Delta t = 1$ second. The burn-in time is to ensure the simulation has reached steady state before changes to the inputs are applied, the burn-in time is 500 seconds. Figure 2 shows one sample of data; here, the $H_{P,init}$ is reduced by 89.03% over a 301.41 second time period.

From inspection, the time step $\Delta t = 1$ second is fine for the dynamical response of the system. Thus, for the purposes of surrogate model training and prediction, the data is coarse grained by a factor of 10. Coarse graining the data means each surrogate model will require less time delays to capture the physical response, reducing the computational cost.

Surrogate Models

We seek a surrogate model to predict the system state at time step $k + 1$ given the system states $\mathbf{x}^{(i)} \in \mathbb{R}^{n_x}$ and system inputs $\mathbf{u}^{(i)} \in \mathbb{R}^{n_u}$ at time steps $i = [k, ..., k - L]$, where L is the number of time delays. Let $\mathbf{\Sigma}$ be the parameters of the surrogate model f, then f can be formulated by the following mathematical representation:

$$\mathbf{x}^{(k+1)} = f(\mathbf{x}^{(k)} \dots \mathbf{x}^{(k-L)}, \mathbf{u}^{(k+1)} \dots \mathbf{u}^{(k-L)}, \mathbf{\Sigma}). \quad (1)$$

The LSTM and ARIMAX surrogate model frameworks are formulated and trained using SAM simulation data to satisfy Equation 1. Each model has its own set of uniquely trainable parameters contained in $\mathbf{\Sigma}_{LSTM}$ and $\mathbf{\Sigma}_{ARIMAX}$.

LSTM The LSTM network structure, shown in Figure 3, consists of the input LSTM layer, N_h hidden LSTM layers, and a dense layer for obtaining the correct output dimensionality. The number of nodes for the input layer and the N_h hidden layers are h_{in} and $h_1, \dots, h_{N_h}$ respectively.

The parameters of the LSTM framework $\mathbf{\Sigma}_{LSTM}$ consists of the weight matrices for each LSTM layer and the dense layer. This is given by the following expression:

$$\mathbf{\Sigma}_{LSTM} = [\sigma_i, \sigma_1, \dots, \sigma_{N_h}, \sigma_D]. \quad (2)$$

To determine the best number of hidden layers and number of nodes per layer, a hyperparamter study using Bayesian Optimization (Jones et al. 1998) is performed on the proposed LSTM framework. The LSTM model's hyperparameters are N_h, h_{in}, and h_j for $j \in [1, N_h]$, these are characteristic to the model and can limit how well it performs. The search space for the hyperparameters are $N_h \in [0, 9]$, $h_{in} \in [5, 200]$, and $h_j \in [5, 200]$. The optimization is run for maximum of 100 trials, the stopping condition being the validation error falling less than 10^{-6}. Each LSTM model is trained for a maximum of 100 epochs using the ADAM optimizer with a learning rate of 10^{-2}. The training and validation data are two sets of data generated from SAM.

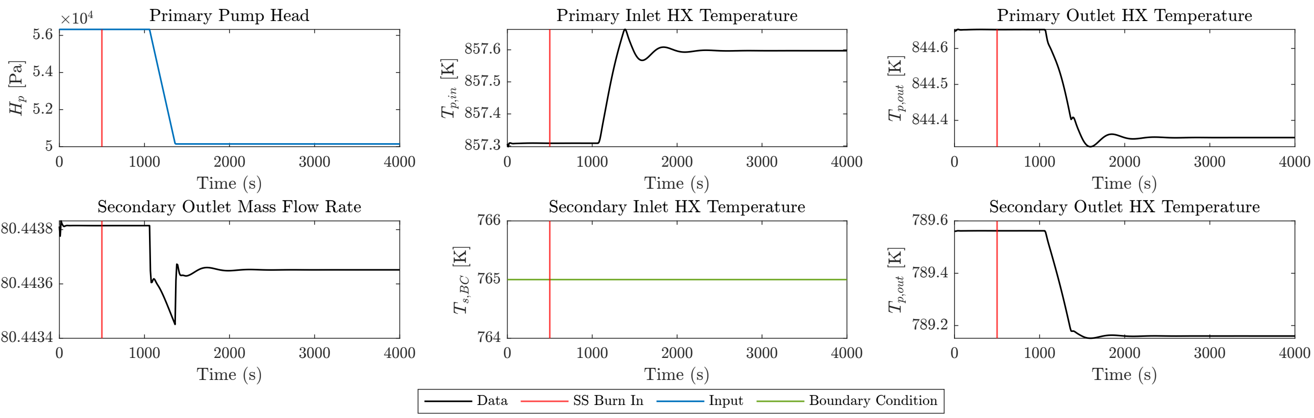

Figure 2: SAM simulation data of FHR model with a ramp change to the primary pump head.

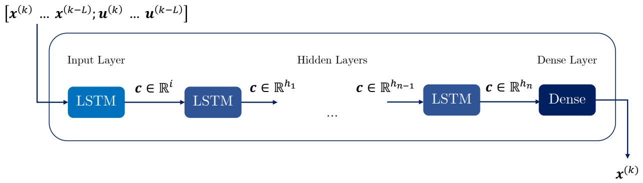
$[\boldsymbol{x}^{(k)}\ \dots\ \boldsymbol{x}^{(k-L)}; \boldsymbol{u}^{(k)}\ \dots\ \boldsymbol{u}^{(k-L)}]$
Input Layer
Hidden Layers
Dense Layer
LSTM
$\boldsymbol{c} \in \mathbb{R}^{i}$
LSTM
$\boldsymbol{c} \in \mathbb{R}^{h_1}$
...
$\boldsymbol{c} \in \mathbb{R}^{h_{n-1}}$
LSTM
$\boldsymbol{c} \in \mathbb{R}^{h_n}$
Dense
$\boldsymbol{x}^{(k)}$

Figure 3: The LSTM framework consists of an LSTM input layer, h_N LSTM hidden layers, and one dense layer.

ARIMAX The ARIMAX(p, d, q, r) is represented by four parameters, the autoregressive polynomial order $p \in \mathbb{Z}_+^{(n_x \times n_n)}$, the integration order $d \in \mathbb{Z}_+$, the moving average polynomial order $q \in \mathbb{Z}_+^{n_x}$, and the extra input polynomial order $r \in \mathbb{Z}_+^{(n_x \times n_u)}$. The number of times the series is integrated, or differenced, is set by the parameter d. For $d = 1$, the ARIMAX model becomes:

$$\Delta \mathbf{x}^{(k)} = c + \sum_{i=1}^{p} \alpha_i \Delta \mathbf{x}^{(k-i)} + \sum_{j=1}^{r} \beta_j \Delta \mathbf{u}^{(k-j)} + \sum_{\ell=0}^{q} \gamma_\ell \epsilon^{(k-\ell)}, \tag{3}$$

where $\Delta \mathbf{x} = \mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}$ and $\Delta \mathbf{u} = \mathbf{u}^{(k)} - \mathbf{u}^{(k-1)}$. The error residual $\epsilon^{(k)} = \mathbf{x}^{(k)} - \tilde{\mathbf{x}}^{(k)}$ is the difference between the true value $\mathbf{x}^{(k)}$ and the predicted value $\tilde{\mathbf{x}}^{(k)}$. Given this model, the parameter matrix $\mathbf{\Sigma}_{ARIMAX}$ contains the weights of each term in Equation 3.

$$\mathbf{\Sigma}_{ARIMAX} = [c, \alpha_1, \dots, \alpha_p, \beta_1, \dots, \beta_r, \gamma_0, \dots, \gamma_q] \tag{4}$$

To train the surrogate model, first the partial autocorrelation and autocorrelation of the data are computed to set the hyperparameters p and q. For the data shown in Figure 2, these are plotted in Figure 4. These values are used as a guiding factor, but are manually tuned to produce results that are physical and have a low reconstruction error.

Performance Metrics

To compare the performance of each surrogate model, we look at the resulting prediction accuracy, the computational cost, and the complexity of each model.

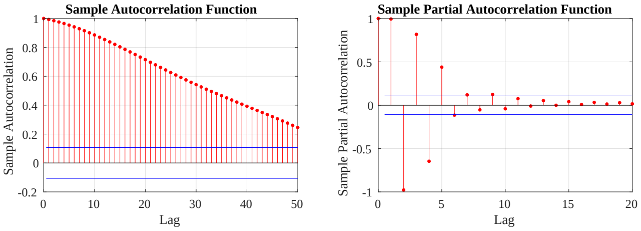

Figure 4: Computed autocorrelation and partial autocorrelation for the data shown in Figure 2.

Accuracy After the surrogate model is trained, it is used to predict the response of the system to an unseen data input. The accuracy is measured by computing the Root Mean Squared Error (RMSE) between the predicted state value $\tilde{\mathbf{x}}_i$ and the true state value $\mathbf{x}_i$. Given a prediction data set with M total observations, the error is computed using Equation 5.

$$RMSE = \sqrt{\frac{1}{M} \sum_{i=1}^{M} (\mathbf{x}_i - \tilde{\mathbf{x}}_i)^2} \tag{5}$$

Computational Time In addition to accuracy, minimizing the model's computational time is vital in order to use the results for real time analysis. First, the total amount of time it takes to train the models is measured in CPU second. Secondly, it is timed for how many CPU seconds it takes to predict one hour of plant simulation data.

Model Complexity Finally, the complexity of each model is measured by the number of trainable parameters. Consider the system model with n_x state values and n_u input values.

For the LSTM framework, the number of trainable parameters $N_{P,LSTM}$ is the total number of weights in the network. The number of weights for the input and hidden layers are given by Equations 6-7; the number of dense layer weights is computed by Equation 8.

$$N_{P,LSTM,in} = 4((n_u + h_1)h_1 + h_1) \tag{6}$$

$$N_{P,LSTM,j} = 4((h_{j-1} + h_j)h_j + h_j) \tag{7}$$

$$N_{P,LSTM,D} = n_x(h_{N_h} + 1) \tag{8}$$

Thus, the total number of LSTM parameters is computed by the following equation:

$$N_{P,LSTM} = N_{P,LSTM,in} + \sum_{j=1}^{N_h} N_{P,LSTM,j} + N_{P,LSTM,D} \tag{9}$$

The ARIMAX number of trainable parameters $N_{P,ARIMAX}$ is the total number polynomial coefficients. This is determined by the following equation:

$$N_{P,ARIMAX} = \underbrace{\sum(p) + n_x^2}_{AR} + \underbrace{\sum(q) + n_x}_{MA} + \underbrace{rn_u}_{X} \tag{10}$$

If $n_x > 1$, then p will be an $(n_x \times n_x)$ matrix, so the total number of parameters is for the autoregressive polynomial is the sum of all matrix values $\sum(p)$ and n_x^2 to account for the constant c in Equation 3. In addition, the number of polynomial coefficients for the moving average terms is the sum of the values in the n_x-dimensional vector q, plus n_x since the error is accounted for in time steps $k-q$ to $k+1$. If $n_u > 1$, the number of coefficients for the input terms will be rn_u

Results

The surrogate models are trained to predict the primary inlet temperature to the heat exchanger $T_{p,in}$ given an input change to the primary pump head H_P. Thus, $n_x = 1$ and $n_u = 1$.

LSTM Hyperparameter Optimization

Using a time-delay of 40, the resulting framework of the LSTM from hyperparameter optimization is listed in Table 1. Here the number of hidden layers N_h and number of neurons for each layer n_i are listed. The training data loss is 0.7052, the validation loss is 4.3609 for the first dataset.

Parameter	Value
N_h	0
n_{in}	90

Table 1: LSTM framework hyperparameters.

ARIMAX Hyperparameters

Using the guidance of the partial autocorrelations and the autocorrections, the surrogate model is ARIMAX($p = 5, d = 1, q = 5, r = 15$). The autocorrelations do not show any significant change for a specific number of lags, and the RMSE of the training reconstruction did not decrease drastically with larger a q or a larger r. So $q = 5$ is selected to match $p = 5$ and $r = 15$ was manually chosen.

Training and Prediction

The results of two datasets are presented. Each dataset contains training and prediction data for each surrogate model to calibrate the trainable parameters and test. For the LSTM training, the iterative training and validation loss is plotted in Figure 5. The performance metrics include the RMSE of the prediction data in addition to the number of trainable parameters for each model. The training time is the CPU time it takes to train surrogate model parameters, and the prediction time of one hour of simulation time is measured on the prediction data.

The Dataset #1 results are found in Figure 6 and Table 2. For this case, both the training data and prediction data contain one ramp change to the input. Dataset #2 results are found in Figure 7 and Table 2. More data is used to train the model, where five sets of ramp changes are panelled after one another. The prediction data also consists of five different ramp changes to the input.

Discussion

Accuracy in Prediction

The ARIMAX provides the most accurate model from visual inspection and the listed error. The LSTM has high errors for the training and prediction data, and does not fully capture the physical nature of the reactor dynamics.

Training LSTMs requires a large database of data. For a digital twin, experimental data collection is limited and computational physics models are expensive. The nuclear reactor is a dynamical system with routine configuration adjustments, thus making large data generation difficult for the LSTM. The ARIMAX was able to reconstruct and predict using only the one set of temporal data. With an increased number of training samples seen in Dataset #2, the reconstruction and prediction performance improved for the LSTM.

Extending this surrogate model to multi-dimensional state prediction with multiple inputs is more tangible with the ARIMAX surrogate. The structure also allows for each state to be dependent its own state values, the multiple inputs, and other state values of the system.

Computational Costs

Comparing the training time of both surrogate models, ARIMAX clearly stands out as the faster option. The ARIMAX model uses a automated method search optimization algorithm to iteratively estimate the parameters. The method rotates between four line search algorithms at each iteration: Gauss-Newton least-squares; Levenberg-Marquardt least squares; adaptive subspace Gauss-Newton; and steepest descent least-squares. The first computed decent direction that results in a reduction of the error is taken. For the LSTM hyperparameter optimization, a Bayesian Optimization algorithm is used. At each iteration, the algorithm builds a Gaussian Progress (GP) Regression model fitted to the collected data. New sets of hyperparameters are sequentially selected according the GP model of the parameter space with respect to the validation error.

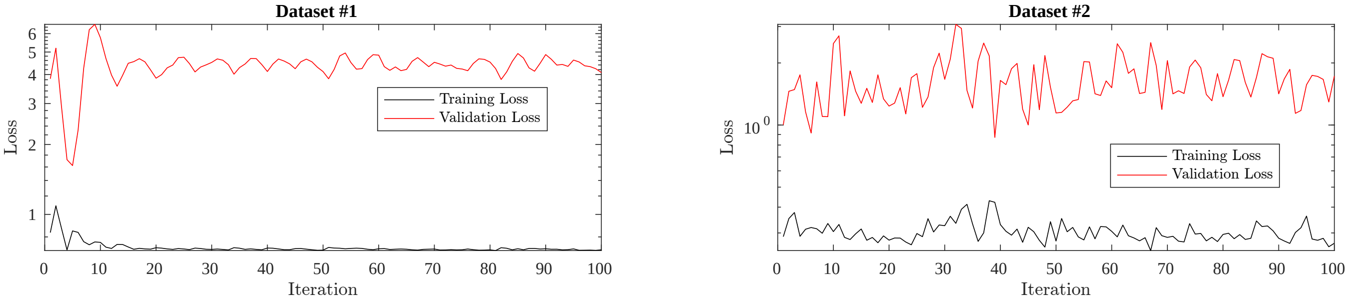

Figure 5: LSTM training and validation loss.

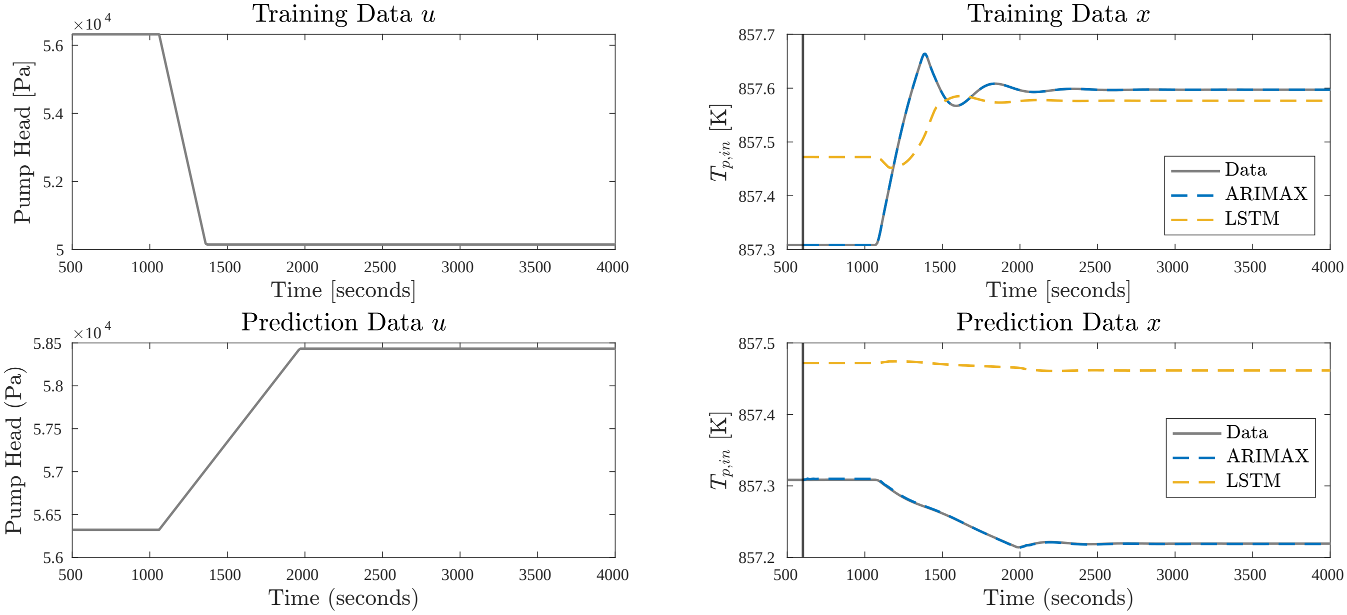

Figure 6: Dataset #1: Training and prediction results.

While the hyperparameter techniques are not the same, the highly notable difference in the number of trainable parameters impacts the surrogate modeling in training. Training the LSTM is a computationally intensive process, determining the thousands of values for the neural network weights requires a long time and cannot be practically implemented if these need to be re-computed often due to new physical asset configurations. This translates to prediction, requiring significantly more floating point operations per second (FLOPS) in order to predict each time step.

Model Complexity

The success of LSTMs in a diverse set of applications have made them a conventional choice for sequential or time-varying data prediction. However, their demands for large sets of training data and long training times are due to a very complex internal structure. Each LSTM layer has four weight matrices, one for the forget and output gates, and two for the input gate. The number of weights is directly related to the number of inputs or outputs, any changes to these values increase the number of weights by a factor of four. These seemingly add up quickly as hidden layers are added. Instead of jumping to a complex LSTM, one should first experiment with models that use basic structures.

The optimized LSTM structure found for the reactor dynamics application needs to train over 40,000 parameters in contrast to the ARIMAX's 22 parameters. With the given data set cases, ARIMAX was able to provide better accuracy than the LSTM. The idea of *parsimony* (Box and Jenkins, 1980) is to have accurate models with the smallest possible number of parameters. This idea discourages the initial jumping to complex models when one should first experiment with models that use simple structures to represent the major trends in the data. With this reactor dynamics application, the change in temperature response of $T_{p,in}$ was simple when the value of the primary pump head is modified. This simple one input, one state model should not require deep neural networks in order to generate reliable prediction results, and less sophisticated models should be tried first.

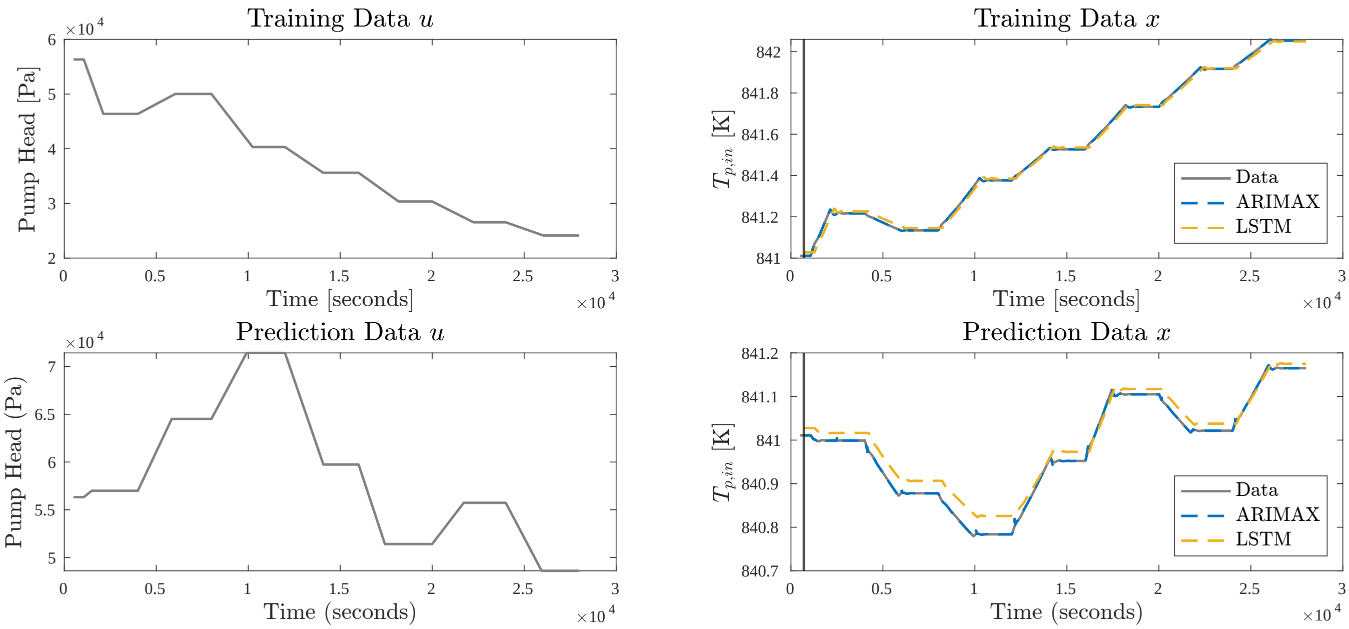

Figure 7: Dataset #2: Training and prediction results.

Dataset	Model	RMSE	Training Time [s]	Prediction Time [s]	N_P
1	LSTM	4.62	1.02×10^1	1.13×10^1	40,051
	ARIMAX	4.60×10^{-3}	1.28	1.45×10^{-1}	22
2	LSTM	3.24×10^1	5.51×10^1	4.61×10^1	40,051
	ARIMAX	1.4×10^{-3}	1.80	2.50×10^{-2}	22

Table 2: Performance metrics for Dataset #1 and Dataset #2.

Conclusion

Both the LSTM and ARIMAX models are suitable surrogate model options for transient dynamical systems. However, the ARIMAX becomes the clear choice when considering the requirements of a digital twin, especially with tight requirements on data sparsity, training time and predictive performance. The LSTM has a high computational cost due to the requirement to train thousands of neural network weights and the amount of training data required. The simplicity of the ARIMAX model allows for fast training and accurate prediction when various reactor configurations need to be tested. In this work, the optimized LSTM structure for the reactor dynamics application needs to train over 40,000 parameters in contrast to the ARIMAX's 22 parameters; and cannot even achieve the same level of accuracy. Indeed, since the experiments in this work were specific to the problem at hand, we do not intend to convey the message that the above conclusions hold for all time series problems. We also acknowledge that LSTMs can be enhanced in a number of ways to serve the specific problem at hand. Nevertheless, we emphasize that complex Deep Neural Networks are not necessarily needed to provide accurate results. Instead, the construction of simple foundational regression models can provide the same or better accuracy with lower computational costs. We intend to scale the ARIMAX model to high-dimensional state estimation and add the quantification of predictive uncertainty. Eventually, the dynamics surrogate model will be embedded into a digital twin.

Acknowledgments

This work was funded by APRA-E under the project *SAFARI: Secure Automation for Advanced Reactor Innovation*.

References

Box, G. E.; and Jenkins, G. M. 1970. *Time Series Analysis: Forecasting and Control*. Holden-Day.

Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Computation*, 9(8): 1735–1780.

Hu, G.; O'Grady, D.; Zou, L.; and Hu, R. 2020. Development of a Reference Model for Molten-Salt-Cooled Pebble-Bed Reactor Using SAM. Technical Report ANL/NSE-20/31, Argonne National Laboratory.

Hu, R.; Zou, L.; Hu, G.; Nunez, D.; Mui, T.; and Fei, T. 2021. SAM Theory Manual. Technical Report ANL/NSE-17/4 Rev. 1, Argonne National Laboratory.

Jones, D. R.; Schonlau, M.; and Welch, W. J. 1998. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13(4): 455–492.

Locatelli, G.; Mancini, M.; and Todeschini, N. 2013. Generation IV nuclear reactors: Current status and future prospects. *Energy Policy*, 61: 1503–1520.

Zhao, H.; Fick, L.; Heald, A.; Zhou, Q.; Richesson, S.; Sutton, N.; and Haugh, B. 2022. Development, verification, and validation of an advanced systems code KP-SAM for Kairos Power fluoride salt-cooled high-temperature reactor (KP-FHR). *Nuclear Science and Engineering*, 1–27.