Samuel M. Hipple
Simulation Modeling and Decision Science
Program Ames Laboratory,
1620 Howe Hall,
Ames, IA 50011
e-mail: smhipple@iastate.edu

Harry Bonilla-Alvarado
Simulation Modeling and Decision Science
Program Ames Laboratory,
1620 Howe Hall,
Ames, IA 50011
e-mail: hbonilla@iastate.edu

Paolo Pezzini
Simulation Modeling and Decision Science
Program Ames Laboratory,
1620 Howe Hall,
Ames, IA 50011
e-mail: ppezzini@ameslab.gov

Lawrence Shadle
National Energy Technology Laboratory,
3610 Collins Ferry Road,
Morgantown, WV 26507
e-mail: Lawrence.Shadle@netl.doe.gov

Kenneth M. Bryden
Simulation Modeling and Decision Science
Program Ames Laboratory,
1620 Howe Hall,
Ames, IA 50011
e-mail: kmbryden@iastate.edu

Using Machine Learning Tools to Predict Compressor Stall

Clean energy has become an increasingly important consideration in today's power systems. As the push for clean energy continues, many coal-fired power plants are being decommissioned in favor of renewable power sources such as wind and solar. However, the intermittent nature of renewables means that dynamic load following traditional power systems is crucial to grid stability. With high flexibility and fast response at a wide range of operating conditions, gas turbine systems are poised to become the main load following component in the power grid. Yet, rapid changes in load can lead to fluid flow instabilities in gas turbine power systems. These instabilities often lead to compressor surge and stall, which are some of the most critical problems facing the safe and efficient operation of compressors in turbomachinery today. Although the topic of compressor surge and stall has been extensively researched, no methods for early prediction have been proven effective. This study explores the utilization of machine learning tools to predict compressor stall. The long short-term memory (LSTM) model, a form of recurrent neural network (RNN), was trained using real compressor stall datasets from a 100 kW recuperated gas turbine power system designed for hybrid configuration. Two variations of the LSTM model, classification and regression, were tested to determine optimal performance. The regression scheme was determined to be the most accurate approach, and a tool for predicting compressor stall was developed using this configuration. Results show that the tool is capable of predicting stalls 5–20 ms before they occur. With a high-speed controller capable of 5 ms time-steps, mitigating action could be taken to prevent compressor stall before it occurs. [DOI: 10.1115/1.4046458]

Keywords: energy conversion/systems, natural gas technology, compressor surge and stall, machine learning, long short-term memory, turbomachinery

1 Introduction

The growing popularity of renewable energy combined with the rising demand for electrical power requires that today's conventional power plants have high efficiency, low emissions, and high flexibility [1,2]. With low cost, high efficiency, and high flexibility, natural gas turbines are well suited to become the main load following component of the power grid. However, high load following flexibility represents one of the most critical problems to be addressed even by natural gas turbines. Forcing traditional fossil generation assets to operate in a load following mode strongly affects the dynamic operation of these systems. For instance, in gas turbine systems compressor instabilities such as surge and stall may become a significant problem.

Compressor stall occurs when the inlet flow is separated from the compressor blade due to an increase in the angle of incidence between the inlet air and the blade [3]. This flow separation results in turbulence along the blade, causing a localized pressure drop. When left unchecked, the turbulent flow propagates throughout the compressor until a complete choking of the airflow occurs [3]. This phenomenon, called surge, results in strong pressure pulsations that can cause extensive damage to the turbomachinery [4]. Although compressor stalls are yet to be fully understood [5], these instabilities are generally a function of the mass flowrate and the pressure ratio between the inlet and the outlet of the compressor. When the mass flowrate decreases relative to the compressor pressure ratio, stall may occur.

Plotting stall inception points as shown in Fig. 1 produces what is referred to as a "surge line." Generally, operating close to the surge line produces higher compressor efficiency but increases the risk of stall. As the push for more efficient power generation continues, compressors in natural gas turbine systems will be required to operate closer to the surge line. This, coupled with highly dynamic load following operations, means that the potential for compressor stall will continue to rise. Because these issues can lead to major turbomachinery failures, several techniques that can detect, control, or predict compressor surge and stall have been studied in the research field.

In the first area of research, compressor surge, and stall detection, Day performed the most notable work because Day was able to detect stall inception by measuring rotor tip velocity at six annular positions in the compressor [6]. However, this method required a compressor that had to be retrofitted with embedded sensors, which is not common in most applications. Although advanced methods for detecting compressor stalls are well researched, no current detection systems can identify precursors to a stall that might indicate its formation in advance [7]. In addition to detection, active control and mitigation of compressor stall represents a second main area of research with many different strategies and methods in use [8]. Three primary methods have been tested and implemented to actively control compressor surge and stall; these include bleed-air (BA) valves, controlled inlet air jets, and variable inlet guide vanes [9–12]. In their review, Gu et al. [9] concluded that inlet air jets and guide vanes are more effective at reducing stall, but bleed valves are effective for both stall and surge.

While stall detection, control, and mitigation have been widely studied, work in compressor stall prediction has been limited mostly to numerical models. The most common is the Moore and Greitzer model [13], which aims to capture the nonlinear characteristics of compressor stall. However, the Moore–Greitzer model

Contributed by the Advanced Energy Systems Division of ASME for publication in the Journal of Energy Resources Technology. Manuscript received November 19, 2019; final manuscript received February 9, 2020; published online February 25, 2020. Assoc. Editor: Ronald Breault.

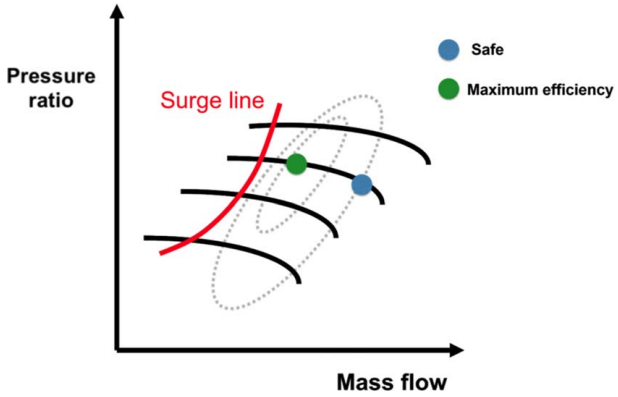

Fig. 1 Operating points on a compressor surge map

assumes incompressible flow within the compressor, which is not valid for high-speed compressors [9]. Physics-based models generally rely on simplifications and assumptions of ideal behavior, which makes them difficult to apply to dynamic events in highly coupled systems such as turbomachinery. For this reason, a different approach is needed to capture the system dynamics that can cause compressor stall so action can be taken to prevent it.

Machine learning algorithms have the ability to process extensive, multidimensional datasets and establish connections between behaviors that cannot be identified in other ways. Once a machine learning algorithm has identified these connections, it can predict outcomes based on provided inputs. Early work has been done in sequence prediction and time series forecasting with methods utilizing machine learning such as support vector machines and neural networks, and more specialized methods quickly emerged such as recurrent neural networks (RNNs) [14,15]. Long short-term memory (LSTM) is a type of recurrent neural network that has been studied for time series prediction and anomaly detection. Malhorta et al. [16] utilized LSTM networks to model normal behavior in several datasets. Using the prediction error distribution from these models, they demonstrated that anomalies could be detected. With its previous success in time series prediction and anomaly detection, machine learning is a viable approach for predicting compressor stall.

In this paper, a prediction tool based on a machine learning algorithm was used to predict compressor stall. The prediction tool was developed using an LSTM neural network where multivariate time series data, acquired at a 5-ms rate from a 100 kW recuperated gas turbine power system designed for hybrid configuration, were used to train the model. The neural network was trained with two approaches, classification and regression, which were compared to understand the performance for predicting stall events. The regression approach was then implemented with a forecasting threshold logic to predict stall inception points. Finally, the application of this tool was evaluated by proposing how it could be combined with the current stall recovery scheme of the micro gas turbine hybrid system.

2 Experimental Facility

Experimental data from the Hybrid Performance (Hyper) facility at the National Energy Technology Laboratory served as a training dataset to develop the compressor stall prediction tool. Hyper consists of a modified recuperated gas turbine and a cyber-physical solid oxide fuel cell (SOFC). The cyber-physical SOFC is a combination of physical and virtual components designed to emulate the dynamic operation of a real SOFC in a SOFC/GT hybrid power system [17]. This makes it possible to conduct studies on system integration and controls without the risk of destroying a fragile and expensive SOFC [17,18]. A diagram depicting the layout of the Hyper facility is shown in Fig. 2. In the direct-fired SOFC/GT, the fuel cell stack and air/fuel distribution components are inserted between the compressor and the turbine. The compressor inlet takes in ambient air and increases its pressure while reducing its volume. Compressed air passes through a recuperated system that further increases the temperature of the air before reaching the fuel cell system. The turbine expands the fluid coming out of the fuel cell system, extracting mechanical power by decreasing pressure and increasing volume.

Turbomachinery failures such as stall may lead to the destruction of the fuel cell, making the prevention and control of this phenomenon crucial for the effective operation of the SOFC/GT. In hybrid systems such as Hyper, the compressor volume can be up to 200 times larger than that observed in a traditional gas turbine cycle [19]. As such, to supply the same mass flowrate as a compressor in a traditional gas turbine configuration, the compressor in the SOFC/GT configuration will operate closer to the surge line due to the larger volume which creates a significant pressure drop to overcome. The increase in volume coupled with the significant pressure drop creates operational and control challenges in the SOFC/GT system. The complexity of stall as well as the sensitivity of

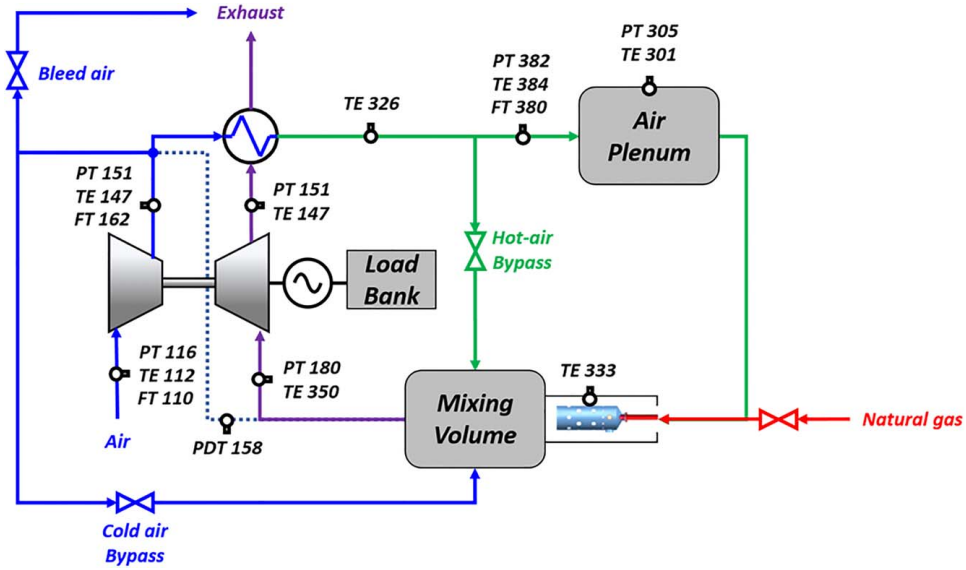

Fig. 2 Diagram of the hyper facility

the fuel cell material augments these operational challenges that make it difficult to establish a real-time model for predicting and preventing stall [10].

2.1 Hardware Components. A 120 kW Garrett Series 85 auxiliary power unit was modified for this testing facility. The gas turbine compressor system is composed of a single shaft, directly coupled turbine, and a two-stage radial compressor. The compressor was designed to deliver approximately 2 kg/s of discharge flow and a pressure ratio of 4:1. The gas turbine is coupled to a gear-driven asynchronous generator, and the load to the generator is applied by a 120 kW resistor bank that dissipates the power output of the source. The load bank is used to replicate real-life demand of a power generation system connected to the grid. The turbine operates at a nominal speed of 40,500 rpm, and a Woodward proportional-integral-derivative (PID) controller, capable of 5-ms time-steps, acts on the swift fuel valve during startup operation or to maintain constant turbine speed during electric load perturbation. The swift fuel valve is electrically actuated; a high-speed stepper motor rotates a 2.54 cm sonic needle and nozzle, which controls the fuel flow going into the combustor at a 5-ms rate if needed.

The BA valve was installed to bypass air from the compressor discharge directly into the atmosphere. The BA valve reduces the total air flow through the turbine and heat exchangers, which is offset by the increase in air flow at the compressor inlet. This valve is also used to ensure enough stall margin to the compressor during startup. A 6-in. valve with a 3-in. body is used as the BA actuator in the Hyper facility. The range of operation is between 100% and 88% of the closing position when the electric load is engaged to the turbine. During startup and nominal condition operations, the valve is set at 94% to guarantee enough stall margin to the compressor. The cold-air (CA) bypass valve is used to bypass cold air from the compressor directly into the turbine inlet through the mixing volume, and the hot-air (HA) bypass valve is used to bypass air from the outlet of the heat exchangers directly into the turbine inlet through the mixing volume.

2.2 Sensors. Various sensors are mounted in the hardware system to measure mass flow, pressure, temperature, and rotational speed. These sensors are generally sampled at a rate of 200 measurements per second. Three optical sensors are used to measure turbine rotational speed. Each sensor optically measures the speed of the light reflected from the rotating target located at the end of the generator shaft. These three signals are transmitted to the control system every 5 ms and averaged to give an average turbine speed. The compressor inlet flow is measured using an annular flow element that provides a mechanical average of the difference between stagnation pressure and static pressure in the inlet pipe to determine the flowrate. Compressor inlet temperature and pressure, measured with a thermocouple and a pressure transducer, determine the ambient condition in the facility test cell during operation and are used to calculate the corrected air flow and compressor pressure ratio.

3 Experimental Operating Conditions

At the Hyper facility, thermal steady-state under nominal conditions is needed before any experiment is performed. The thermal steady-state condition is achieved when the turbine rotational speed reaches the nominal setpoint of 40,500 rpm, and the skin temperature of the mixing volume varies less than 1.0 K for a 30 s period. When the nominal turbine speed is reached, the electrical load bank is generally used to engage a 40 kW resistive load to dissipate the energy produced by the turbine generator. During startup and around nominal conditions, a single-input single-output PID controller maintains nominal turbine speed by handling the fuel flow into the gas turbine combustor. An initial scoping study was performed to determine the operating conditions in which incipient stalls were observed without completely surging the compressor. This test started at nominal conditions and the BA valve and CA and HA bypass valves were used to reduce the stall margin by restricting the amount of air flow in the compressor. First, the BA valve was closed from a 6% to a 0% open position. Next, the CA bypass valve was closed from a 40% to a 28% open position while HA bypass valve was maintained at a 25% open position.

After the operating conditions were determined, four tests, which are summarized in Table 1, were conducted to apply a more systematic approach to reducing the surge margin. These four datasets were obtained under the same operating conditions, but the tests were conducted on different dates. Therefore, some inlet conditions on the compressor may differ, but the same test procedure was used for each experiment. In case 1, the BA valve was set at the fully closed position, the CA bypass valve at 28% open, and the HA bypass valve at 25% open. This dataset included 7692 data points of operation near the surge line with two incipient stall events observed. In case 2, the BA valve was set at 6% open, the CA bypass valve at 40% open, and the HA bypass valve at 25% open. This dataset included 9216 data points of nominal operating conditions, and no stall events were observed. In case 3, the BA valve was closed from 6% open to 0% open in one-step change, the HA bypass valve was maintained at 25% open, and the CA bypass valve was changed from 40% to 30% open in 5% step changes, then to 28% open in one step change. The system was allowed to stabilize for 50 s between each step change. This dataset included 56,960 data points with the facility operating from nominal conditions to near the surge line. As shown in Fig. 3, the third test was divided into two datasets; approximately 80% of the data were used for training the model (case 3.1) and the remaining data were used for validation (case 3.2). Case 3.1 included eight incipient stall events, and case 3.2 included seven incipient stall events with some stall propagating and becoming

Table 1 Operating conditions for each test case

Test case	Data points	Incipient stall events	Operating conditions	Use in machine learning model
1	7692	2	Near surge line	Training
2	9216	0	Nominal	Training
3.1	46,080	8	Nominal to near surge line	Training
3.2	10,880	7	Near surge line	Validation

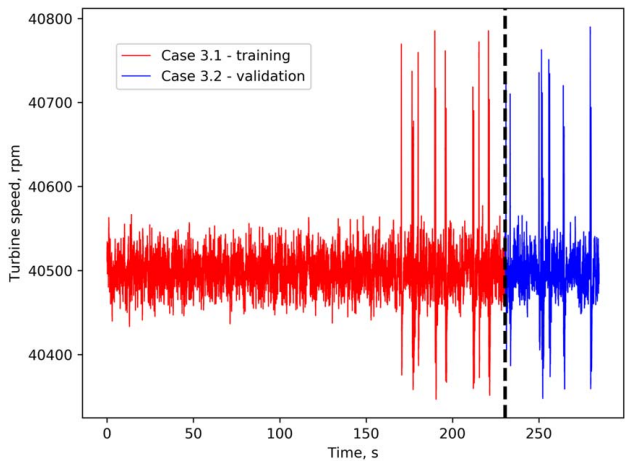

Fig. 3 Division of test case 3

surge but not enough to terminate the experiments. These experiments were short due to the potential for mechanical damage to the facility and turbomachinery components. In total, 62,988 data points were used to train the machine learning model and 10,880 data points were used to validate the model.

4 Machine Learning Model

When considering the machine learning tool to utilize for this study, several characteristics of the data were evaluated including magnitude, type, number of variables, noise, and complexity. Based on the characteristics of the data and given that the objective of this study involves sequence prediction, a RNN was an obvious candidate. An RNN is a form of neural network that utilizes feedback to impose a "memory" in each network node. A simplified diagram of an RNN is shown in Fig. 4. Generic RNNs are useful for pattern recognition but are limited to short-term applications due to unstable backflow error [20]. Introduced by Hochreiter and Schmidhuber [20], the gradient-based LSTM learning algorithm is a form of RNN that was developed to overcome backpropagation error instability and can be used for extended sequence prediction.

The LSTM learning algorithm utilizes memory cells with logical gates to enforce stable error flow. As shown in Fig. 5, a memory cell contains several activation functions. Each activation function uses logical gates to determine the output of the function. The gates utilize logistic sigmoid (σ) and hyperbolic tangent functions (tanh) to scale inputs and previous values. The weights for each gate (W_g, V_g) are set at each time-step by the learning algorithm according to the error gradient. These weights collectively form the machine learning model. The forget gate activation function (f_t), defined by Eq. (1), looks at some input x_t and regulates what information from the previous hidden state (h_{t-1}) is "forgotten" by the cell

$$f_t = \sigma(W_f h_{t-1} + V_f x_t) \tag{1}$$

The input gate activation function (i_t), defined by Eq. (2), regulates what information from the input array is important to the cell. The same input parameters (h_{t-1}, x_t) of Eq. (1) are fed into Eq. (2) and similar to Eq. (1) the weights W_i and V_i forms the output i_t

$$i_t = \sigma(W_i h_{t-1} + V_i x_t) \tag{2}$$

The candidate cell state (c_t), defined by Eq. (3), creates the new values that are possible for the cell's output state. Similar to Eqs. (1) and (2), h_{t-1} and x_t represent the output from the previous hidden state and the data point, respectively. In this case, W_c and V_c represents the weights for the candidate cell state c_t

$$c_t = \tanh(W_c h_{t-1} + V_c x_t) \tag{3}$$

The output gate activation function (o_t), defined by Eq. (4), regulates what information is passed along as an output for the cell. Similar to the previous equations, W_o and V_o are the weights determined at each time-step by the error gradient machine learning algorithm, and h_{t-1} and x_t are the hidden state and the data point

$$o_t = \sigma(W_o h_{t-1} + V_o x_t) \tag{4}$$

The current cell state (C_t), defined by Eq. (5), combines what should be remembered from the previous cell state with new values that have been deemed important. The current cell state is defined as a linear combination between the current cell state from the previous state (C_{t-1}), the output of the forget activation function (f_t) at the current state, the output of the current cell state (C_t), and the output of the input gate activation function (i_t)

$$C_t = f_t C_{t-1} + i_t c_t \tag{5}$$

The hidden state (h_t), defined by Eq. (6), normalizes the cell's current state and determines what information actually produced as

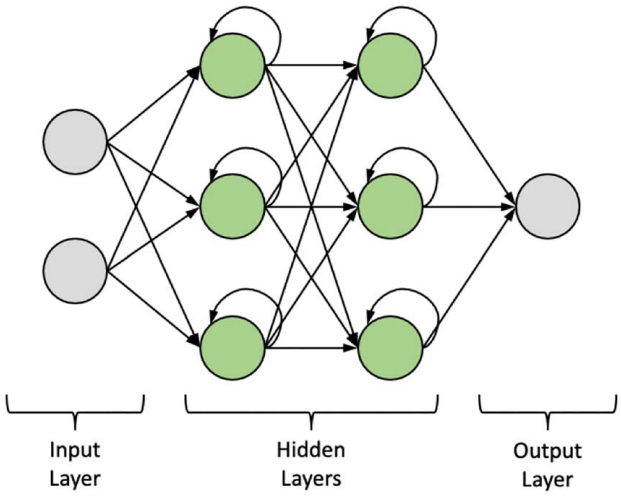

Fig. 4 Example of a recurrent neural network

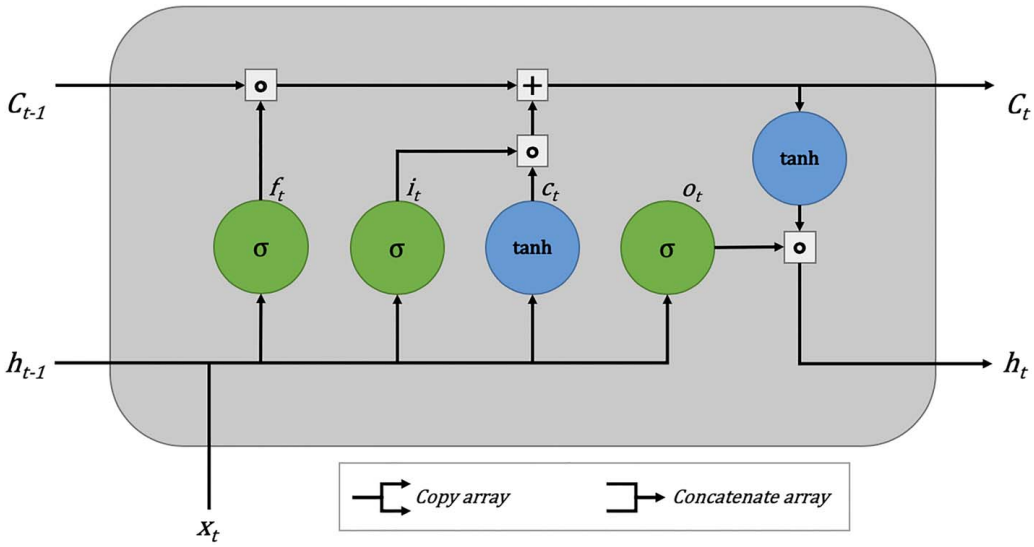

Fig. 5 Memory cell in an LSTM network

an output for the cell. The output gate activation function (o_t) multiplied by the hyperbolic tangent functions (tanh) related to the current cell state determines the hidden state

$$h_t = o_t \tanh(C_t) \tag{6}$$

Unlike other neural networks, LSTM is capable of recognizing patterns over gaps exceeding 1000 steps and is well suited for noisy data, making it the ideal model for this study.

5 Machine Learning Model Setup

Two LSTM model implementations, classification and regression, were studied to determine the best approach. The classification approach requires training data to be manually classified prior to training. The output of a classification model is then a predicted classification. In this study, the training data for the classification approach was composed of 33 sensors as well as manually classified stalls (1 for stall and 0 nonstall). The regression approach, alternatively, predicts the output of a variable based on given inputs. For this test, the sensor data was used as inputs to the LSTM model, which was used to predict turbine speed. Because a stall can be characterized by a large spike in turbine speed, a threshold logic was designed to identify when rpm surpassed a designated stall limit. Steady operation in this test was 40,500 rpm, so the stall thresholds were set at 40,400 rpm and 40,600 rpm (i.e., ±100 rpm). This allowed turbine speed spikes to be used as an indicator that a stall had occurred.

The model was implemented in PYTHON using Keras [21], a simplified interface for neural networks that runs on top of Tensorflow. The Keras interface allows model parameters, referred to as hyperparameters, to be set at a higher level than a manual implementation of LSTM. This enables a more efficient model development process and improved testing capability for a range of model configurations. The first layer of the neural network, the LSTM input layer, was set for the number of inputs fed to the model. The input layer was then followed by another LSTM layer and two fully connected dense layers, which compute the output of the model based on the weights and hidden state. The output layer was set based on the desired output and number of time-steps to predict into the future. The dropout regularization method was implemented in the LSTM layers, which forces the model to exclude a given percentage of inputs and recurrent connections during a memory cell's training. This helps to prevent a highly noise-influenced model (i.e., overfitting) and improves model performance.

Three datasets were used to train the model, ranging from 7692 to 46,080 time-steps in length. Because the datasets were not sequential, the LSTM model was fitted for a given number of epochs on each dataset before moving onto the next. The number of epochs is an LSTM hyperparameter that defines the number of times the learning algorithm iterates over each dataset during the training process. During training, each dataset was separated into smaller batches by Keras. The size of these batches is defined by the user, and the LSTM model weights are updated after a batch of data has processed. The number of input time-steps the learning algorithm trained on was a critical hyperparameter that allowed the model to process enough data to make a good prediction for the defined number of output time-steps.

As previously described, test case 3.2 was used to test the accuracy of the trained model. This dataset was used as inputs for the LSTM model, which predicted the stall classification in the classification approach and the turbine speed in the regression approach. Once the LSTM model output was predicted, stall or normal operating conditions were determined based on the predicted values. This was then compared with the stall results from the original data to determine the model's accuracy for forecasting stall using Eqs. (7)–(10)

$$\text{Accuracy} = \frac{1}{P}(T_P + T_N) \tag{7}$$

$$\text{Error rate} = \frac{1}{P}(F_P + F_N) \tag{8}$$

$$\text{Sensitivity} = \frac{T_P}{T_P + F_N} \tag{9}$$

$$\text{Specificity} = \frac{T_N}{F_P + T_N} \tag{10}$$

5.1 Hyperparameter Tuning and Performance Validation in the Classification Model. An LSTM model structure was trained based on training data sets with labeled data that identify the stall event. The data was labeled using a boolean classification (1 for a stall and 0 for a nonstall) by manually identifying points in the training data when turbine speed passed the designated thresholds (i.e., ±100 rpm around the turbine speed nominal operation). As previously mentioned, to optimize the model, several configurations of hyperparameters were tested, which included the number of epochs, the batch size for each epoch, the number of input time-steps, the number of prediction output time-steps ahead, the size of the hidden layer, and the size of the second hidden layer. From these tests, the batch size for each epoch was set to 30 data points, the best number of input time-steps for the classification model was determined to be 40 time-steps, the optimal number of output time-steps was 10 time-steps, the first hidden layer size was 64, and the second hidden layer size was 128.

After training, the model was validated with the test data set and used to predict a final case with multiple stall events. Accuracy and mean-squared error (MSE) were used to determine the performance of the model for predicting stalls. In this study, the number of training epochs, or iterations over the dataset, were varied while keeping the other hyperparameters constant. The results of the classification tests are shown in Table 2. An average accuracy of 86% was achieved with the classification model. The 15-epoch model with no dropout produced the lowest MSE at 0.00959. During the optimization process, both the test and training had unstable losses, meaning model error did not decrease smoothly as training and testing progressed. Although this did not contribute much to the accuracy of the stall classification, it did affect the MSE. Unstable loss can be an indicator of overfitting, so a dropout layer was considered to stabilize losses. As shown in Table 2, however, the 15-epoch model with 20% dropout produced similar results as the other tests with no dropout.

Table 2 Results of classification tests

Hyperparameters							Results	
Training epochs	Batch size	Number of input time-steps	Number of output time-steps ahead	First hidden layer size	Second hidden layer size	Dropout	MSE	Accuracy, %
5	30	40	10	64	128	0%	0.024577	86.35
10	30	40	10	64	128	0%	0.013339	86.51
15	30	40	10	64	128	0%	0.00959	86.24
15	30	40	10	64	128	20%	0.010293	86.15

5.2 Hyperparameter Tuning and Performance Validation in the Regression Model. The regression model was optimized by changing various hyperparameters, similar to the classification approach. In addition to accuracy, three other metrics were considered: error rate, sensitivity, and specificity. These metrics help to determine what areas the model is strong or weak in predicting stall. Based on the performance comparison, the optimal LSTM hyperparameter configuration had an output sequence of 15 time-steps ahead. The results from this model are shown in Fig. 6. The regression model prediction accuracy was high; however, the sensitivity was low in all hyperparameter configurations, as shown in Table 3. It can be deduced from the high accuracy and low sensitivity that the model performed well in predicting points where stall did not occur, but did not perform well in predicting points where stall did occur. While accuracy was higher than that of the classification model, the poor results for sensitivity showed that further work was required to accurately predict stalls.

6 LSTM Regression Model Implementation

Comparing Tables 2 and 3, which represent the performance of the classification LSTM model against the regression model, it was determined that a higher prediction accuracy was achieved with the regression model. As previously discussed, however, the regression model did not perform well in predicting all data points where the turbine speed surpassed the stall threshold. For the scope of detecting and controlling compressor instabilities, the full dynamics of a stall event must be understood. However, model training shows that the highly variable nature of compressor instabilities makes predicting the dynamics of a stall difficult, even for a machine learning algorithm. The wide variety of existing stall detection, mitigation, and control techniques further complicate stall dynamics and magnify this challenge. The goal of this study, however, is to *predict* compressor stall rather than *detect* it. Although the turbine speed prediction during a stall was not accurate, the predictions at the beginning of each stall event appear to be more accurate. If the onset of a stall can be predicted, then appropriate action can be taken to prevent the stall. The key function of a stall prediction tool, therefore, is to predict the point where a stall begins. Thus, the regression approach was selected as the best LSTM implementation and was combined with a threshold logic that was designed to identify stall inception points.

6.1 Stall Inception Point Identification. A stall is characterized by a large, rapid change in rpm compared with normal operations. In the regression LSTM model testing, this was captured using a turbine speed threshold to detect stall in the prediction data. The stall begins at the moment the turbine speed begins to spike, but due to noise in the turbine speed data, the turbine speed threshold must be set well above normal operation conditions to avoid a false prediction. This means that a delay can occur between the beginning of a stall and when the stall threshold is triggered. Because the rpm changes rapidly, a stall can also be classified using the differential of turbine speed, where an increase in the differential can be used to predict a stall event. Furthermore, the stall inception point can be identified as the moment when the turbine speed differential exceeds a designated threshold. The differential-based threshold approach can detect stall earlier than using a threshold approach only on the turbine speed. This provides an important advantage because a corrective system action could be taken sooner and the chance of avoiding the stall would improve.

The immediate result of a stall may be either an increase or decrease in turbine speed (i.e., a positive or negative differential), so the absolute value of the differential was used to account for both cases with a single threshold. To ensure that the tool can be used for real-time stall prediction, the instantaneous differential was calculated over the turbine speed array using only current and previous data points, as shown in Eq. (11)

$$\frac{\partial \boldsymbol{\omega}}{\partial \boldsymbol{t}}[i] = \frac{\boldsymbol{\omega}[i] - \boldsymbol{\omega}[i-n]}{\boldsymbol{t}[i] - \boldsymbol{t}[i-n]} \tag{11}$$

To reduce noise and better identify outliers (i.e., stall points), the turbine speed data were processed using an unweighted moving

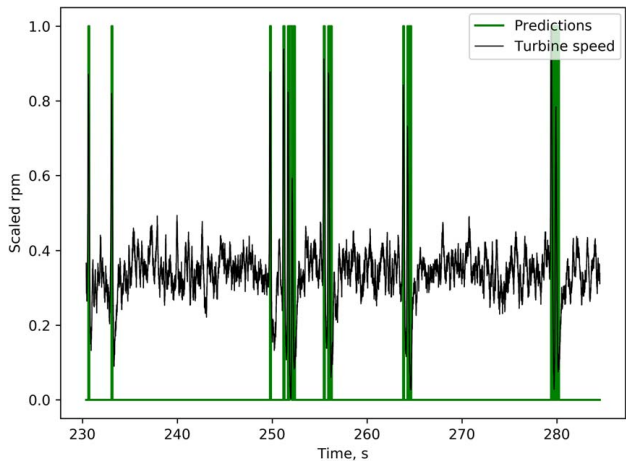

Fig. 6 Regression model results

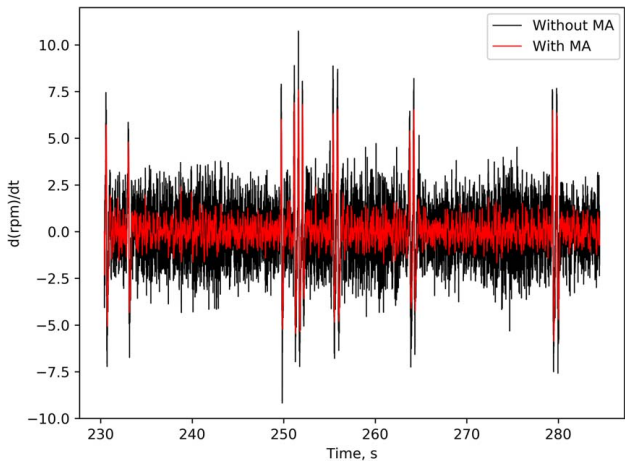

Fig. 7 Effect of moving average on noise and outliers

Table 3 Results of regression tests with various hyperparameter configurations

Hyperparameters							Results			
Training epochs	Batch size	Input time-steps	Output time-steps	Input layer size	Second layer size	Dense layer size	Error rate	Accuracy, %	Sensitivity	Specificity
2	30	40	15	32	64	64	0.050195	94.9805	0.011654	0.998284
5	30	40	15	32	64	64	0.051080	94.8920	0.021617	0.996816
10	30	40	15	32	64	64	0.050429	94.9571	0.013409	0.997947
2	30	40	15	32	64	16	0.050729	94.9271	0.022744	0.997126

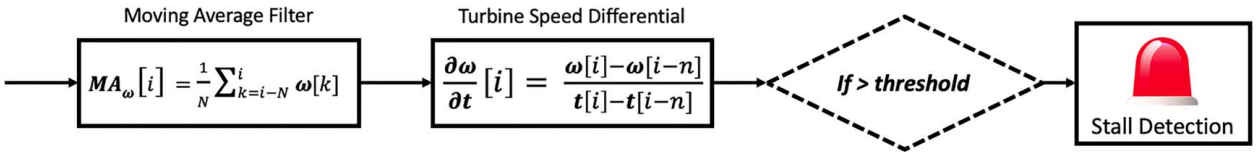

Fig. 8 Stall detection threshold logic

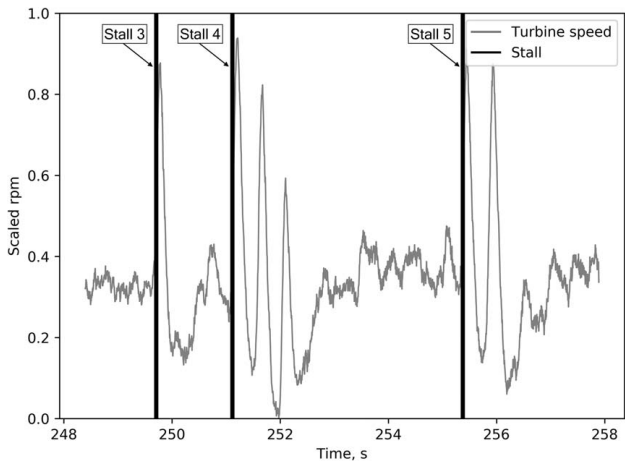

Fig. 9 Sample of turbine speed data during stalls 3–5

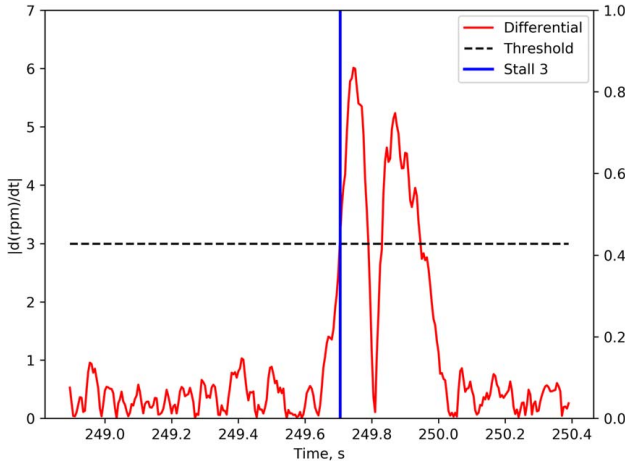

Fig. 10 Stall 3 inception point identification

average filter (Eq. (12)) before taking the differential. Figure 7 shows the turbine speed differential with and without the moving average filter. A window size of 10 data points (0.05 s) for the moving average filter was selected as the optimal setting for the data in this study. The overall threshold logic is shown in Fig. 8.

$$MA_\omega[i] = \frac{1}{N}\sum_{k=i-N}^{i} \omega[k] \tag{12}$$

Figure 9 shows a snippet of turbine speed data in which a stall occurs. It can be seen from these data that a stall is accompanied by several large swings in turbine speed. Using the threshold logic previously described, these swings would cause the turbine speed differential to cross the threshold several times in rapid succession during a single stall event. The objective, however, is to identify only the first point where threshold is crossed in a stall. To avoid this problem, a memory logic was implemented that does not allow a threshold crossing to be identified as a stall inception point within a 200 ms window of the previous threshold crossing. This time period ensured that only one inception point was identified for each stall, but different individual stalls could still be recognized. Figure 10 shows a subsection of test data that has been processed using the threshold to identify a stall point. It can be seen that although the turbine speed differential crossed the threshold several times, the memory logic allowed only the first threshold crossing point in the stall to be identified as a stall inception point.

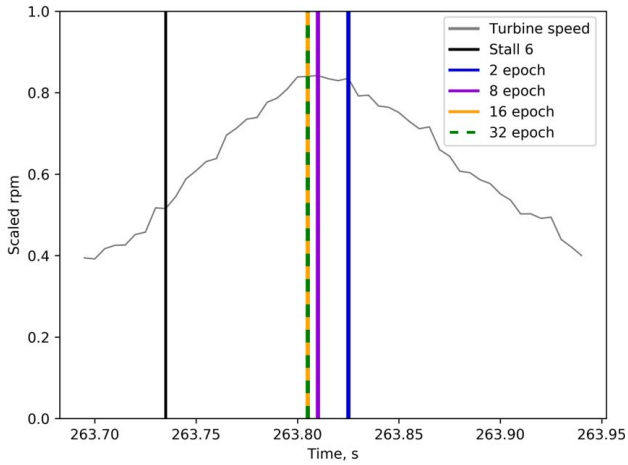

Fig. 11 Stall 6 inception point predictions for various training epochs

7 Results

The stall inception points presented in Fig. 9 (i.e., test case 3.2) were compared for prediction accuracy against the LSTM prediction test data. To align the data for proper comparison, any false predictions were identified manually. Also, the stall points were compared numerically to determine time disparities between real

Table 4 Results of prediction tool with various training epochs

	Prediction error (ms)								
Training epochs	Stall 1	Stall 2	Stall 3	Stall 4	Stall 5	Stall 6	Stall 7	Missed stalls	False stalls
2	70	80	70	65	65	90	60	0	0
8	60	80	65	65	60	75	55	0	0
16	60	70	65	65	60	70	55	0	0
32	65	70	70	–	65	70	60	1	2

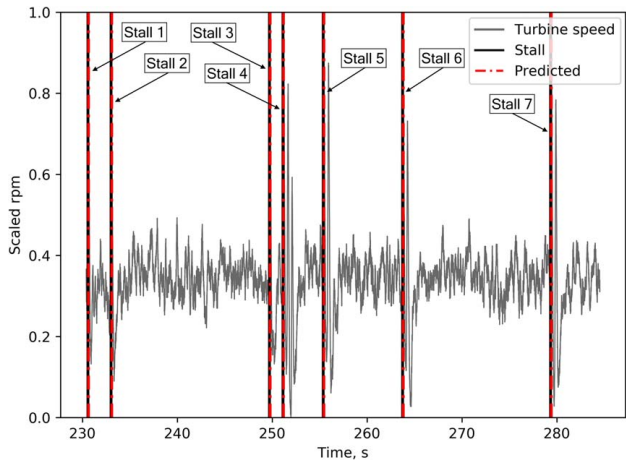

Fig. 12 Comparison of predicted and actual stall inception points for 16 epoch test

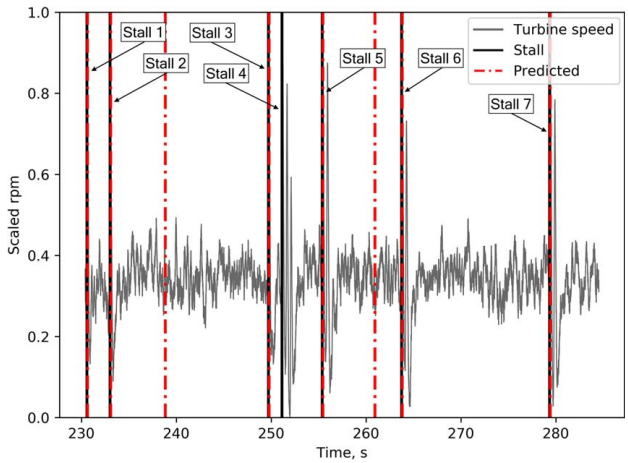

Fig. 13 Comparison of predicted and actual stall inception points for 32 epoch test

and predicted stalls. Most of the optimized hyperparameters presented in Table 3 were used in the test data set shown in Fig. 9. For instance, the number of training epochs used in Fig. 9 was different from Table 3, as shown in Table 4 they were varied and compared to determine optimal settings. In all cases, the turbine speed was predicted 15 time-steps (75 ms) ahead. The results of these tests are shown in Table 4. In the test data set 3.2, seven stall events occur. To evaluate the models, the predicted stall points were subtracted from the real stall points to determine the prediction error in milliseconds.

As shown in Table 4, a general decreasing trend in prediction error was observed from the 2-epoch test case to the 16-epoch test case. This trend is exemplified in Fig. 11, which shows a detailed view of stall 6 and the corresponding predicted stall points for each training epoch configuration. The best results were achieved in the 16-epoch test case. Figure 12 shows the results of this test, where dashed vertical lines represent predicted stall points and solid vertical lines represent actual stall points. It can be seen from the overlap of predicted and actual stall points that all stalls were predicted and no false stall predictions occurred. Prediction errors varied from 55 to 70 ms, meaning the stalls were forecasted to occur later than those in the test data. Because the prediction occurs 75 ms in advance, 5–20 ms remain before the stall occurs for system intervention.

When the model was trained with more than 16 epochs, predictions became less accurate. In the 32-epoch test case, the model gave two false predictions of stall that did not exist in the test data, and also failed to predict stall 4. This is shown in Fig. 13, where two false predictions occur near times 238 s and 261 s, and stall 4 is not predicted at all. The latter is exemplified in Fig. 14, where a detailed view of stalls 3 and 4 shows that the 16-epoch model predicted both stalls, while the 32-epoch model did not predict stall 4.

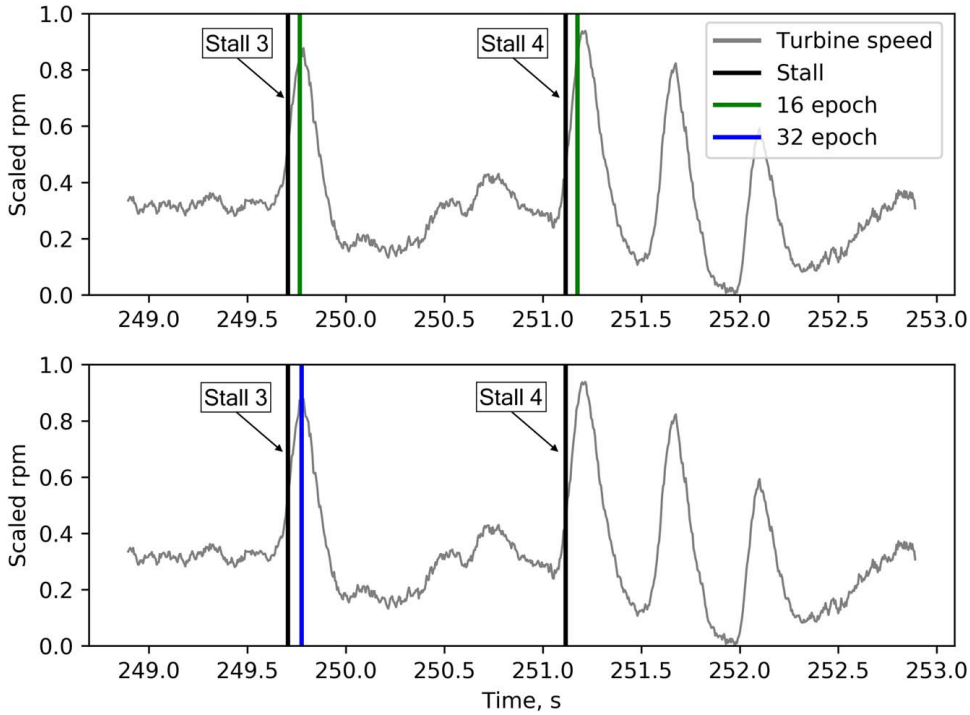

Fig. 14 Comparison of 16 and 32 epoch predictions for stalls 3 and 4 in the test data

8 Conclusion

This paper discusses various techniques for applying machine learning to predict compressor stall in a 100 kW recuperated gas turbine power system designed for hybrid configuration. It was determined that a LSTM neural network was the best model to utilize. Testing concluded that because stalls occur very infrequently relative to nonstalls, a more robust and accurate model could be created using regression to predict turbine speed rather than classification to predict the stalls themselves. The highest accuracies were achieved using the LSTM model to predict turbine speed 15 time-steps (75 ms) ahead with 16 training epochs. This study concludes that by analyzing data with the stall prediction tool, it is possible to predict compressor stalls 5–20 ms before they occur. With a high-speed data acquisition system, this leaves time for preventative action, such as valve changes, to be taken. Specifically, automated control actions could be implemented in combination with the compressor stall prediction model to avoid surge during the operation of a gas turbine system. Sensors from a gas turbine system can be fed to the stall prediction tool, which is capable of predicting a stall event 5–20 ms before it begins. After the 5 ms calculation time-step, an automated controller can open an actuator that will reduce the pressure ratio across the compressor and increase inlet flow. Subsequently, the surge margin will increase and the stall will be prevented. This study shows promise for a new generation of tools that utilize machine learning to predict compressor stall.

9 Future Work

Future work for this project will include a comprehensive study of LSTM hyperparameter configurations to attempt to further extend model prediction times. Once this work is completed, the prediction tool will be implemented on the Hyper system to determine the effectiveness of early intervention on stall reduction and prevention using the Hyper system time scale. Other interesting work would include training the LSTM model on additional datasets, including supplementary sensors or valves in the training data and predicting other parameters beyond turbine speed.

Nomenclature

i = array index
n = number of time-steps to include in differential calculation
N = moving average window size
P = total number of model predictions
c_t = candidate cell state
f_t = forget gate activation function
h_t = hidden state
i_t = input gate activation function
o_t = output gate activation function
C_t = current cell state
F_N = number of stall time-steps predicted as nonstall
F_P = number of nonstall time-steps predicted as stall
T_N = number of nonstall time-steps predicted as nonstall
T_P = number of stall time-steps predicted as stall
V_g = weight of previous state
W_g = weight of hidden state
MA_ω = moving average array for turbine speed
σ = sigmoid squashing function
$\boldsymbol{\omega}$ = turbine speed array
$\frac{\partial \omega}{\partial t}$ = turbine speed differential array

References

[1] Fell, H., and Kaffine, D. T., 2018, "The Fall of Coal: Joint Impacts of Fuel Prices and Renewables on Generation and Emissions," Am. Econ. J.: Econ. Policy, **10**(2), pp. 90–116.
[2] Aghaei, J., and Alizadeh, M.-I., 2013, "Demand Response in Smart Electricity Grids Equipped With Renewable Energy Sources: A Review," Renewable Sustainable Energy Rev., **18**, pp. 64–72.
[3] Tavakoli, S., Griffin, I., and Fleming, P., 2004, "An Overview of Compressor Instabilities: Basic Concepts and Control," IFAC Proceedings Volumes, **37**(6), pp. 523–528.
[4] Almasi, A., 2012, "Latest Techniques and Practical Notes on Anti-Surge Systems for Centrifugal Compressors," Aust. J. Mech. Eng., **10**(1), pp. 81–90.
[5] De Jager, B., 1995, "Rotating Stall and Surge Control: A Survey," Proceedings of 1995 34th IEEE Conference on Decision and Control, New Orleans, LA, Dec. 13–15, IEEE, pp. 1857–1862.
[6] Day, I. J., 1991, "Active Suppression of Rotating Stall and Surge in Axial Compressors," ASME 1991 International Gas Turbine and Aeroengine Congress and Exposition, Orlando, FL, June 3–6, p. V001T01A035.
[7] Tan, C. S., Day, I., Morris, S., and Wadia, A., 2010, "Spike-Type Compressor Stall Inception, Detection, and Control," Annu. Rev. Fluid Mech., **42**(1), pp. 275–300.
[8] Day, I. J., 2016, "Stall, Surge, and 75 Years of Research," ASME J. Turbomach., **138**(1), p. 011001.
[9] Gu, G., Sparks, A., and Banda, S. S., 1999, "An Overview of Rotating Stall and Surge Control for Axial Flow Compressors," IEEE Trans. Control Syst. Technol., **7**(6), pp. 639–647.
[10] Pezzini, P., Tucker, D., and Traverso, A., 2013, "Avoiding Compressor Surge During Emergency Shutdown Hybrid Turbine Systems," ASME J. Eng. Gas Turbines Power, **135**(10), p. 102602.
[11] Weigl, H. J., Paduano, J. D., Frechette, L. G., Epstein, A. H., Greitzer, E. M., Bright, M. M., and Strazisar, A. J., 1997, "Active Stabilization of Rotating Stall and Surge in a Transonic Single Stage Axial Compressor," ASME 1997 International Gas Turbine and Aeroengine Congress and Exhibition, Orlando, FL, June 2–5, American Society of Mechanical Engineers, p. V004T15A034.
[12] Paduano, J., Epstein, A. H., Valavani, L., Longley, J. P., Greitzer, E. M., and Guenette, G. R., 1991, "Active Control of Rotating Stall in a Low Speed Axial Compressor," ASME 1991 International Gas Turbine and Aeroengine Congress and Exposition, Orlando, FL, June 3–6, American Society of Mechanical Engineers, p. V001T01A036.
[13] Moore, F. K., and Greitzer, E. M., 1986, "A Theory of Post-Stall Transients in Axial Compression Systems: Part I—Development of Equations," ASME J. Eng. Gas Turbines Power, **108**(1), pp. 68–76.
[14] Müller, K.-R., Smola, A. J., Rätsch, G., Schölkopf, B., Kohlmorgen, J., and Vapnik, V., 1997, "Predicting Time Series With Support Vector Machines," International Conference on Artificial Neural Networks, Lausanne, Switzerland, Oct. 8–10, Springer, Berlin, Heidelberg, pp. 999–1004.
[15] Kolarik, T., and Rudorfer, G., 1994, "Time Series Forecasting Using Neural Networks," ACM SIGAPL APL Quote Quad, **25**(1), pp. 86–94.
[16] Malhotra, P., Vig, L., Shroff, G., and Agarwal, P., 2015, "Long Short Term Memory Networks for Anomaly Detection in Time Series," Proceedings, Bruges, Belgium, Apr. 22–24, p. 89.
[17] Tucker, D., Pezzini, P., and Bryden, K. M., 2018, "Cyber-Physical Systems: A New Paradigm for Energy Technology Development," ASME 2018 Power Conference Collocated With the ASME 2018 12th International Conference on Energy Sustainability and the ASME 2018 Nuclear Forum, Lake Buena Vista, FL, June 24–28, American Society of Mechanical Engineers, p. V001T04A001.
[18] Pezzini, P., Bryden, K. M., and Tucker, D., 2018, "Multicoordination Control Strategy Performance in Hybrid Power Systems," ASME J. Electrochem. Energy Convers. Storage, **15**(3), p. 031007.
[19] Tucker, D., Shadle, L., and Harun, N. F., 2017, "Automated Compressor Surge Recovery With Cold Air Bypass in Gas Turbine Based Hybrid Systems," International Symposium on Transport Phenomena and Dynamics of Rotating Machinery, Maui, HI, Dec. 16–21.
[20] Hochreiter, S., and Schmidhuber, J., 1997, "Long Short-Term Memory," Neural Comput., **9**(8), pp. 1735–1780.
[21] Chollet, F., 2015, Keras, https://keras.io