Knowledge-Guided Learning of Temporal Dynamics and its Application to Gas Turbines

Pawel Bielski
Karlsruhe Institute of Technology
Karlsruhe, Germany
pawel.bielski@kit.edu

Aleksandr Eismont
Karlsruhe Institute of Technology
Karlsruhe, Germany
ukqln@student.kit.edu

Jakob Bach
Karlsruhe Institute of Technology
Karlsruhe, Germany
jakob.bach@kit.edu

Florian Leiser
Karlsruhe Institute of Technology
Karlsruhe, Germany
florian.leiser@kit.edu

Dustin Kottonau
Karlsruhe Institute of Technology
Karlsruhe, Germany
dustin.kottonau@kit.edu

Klemens Böhm
Karlsruhe Institute of Technology
Karlsruhe, Germany
klemens.boehm@kit.edu

ABSTRACT

Modeling dynamical systems is a fundamental task in scientific and engineering fields, often accomplished by applying theory-based models with mathematical equations. Yet, in cases where these equations cannot be established or parameterized properly, theory-based models are not applicable. Instead, a viable alternative is to learn the system dynamics directly from data, for example with deep learning models. However, traditional deep learning models often produce physically inconsistent results and struggle to generalize to unseen data, especially when training data is limited. One solution to this shortcoming is knowledge-guided deep learning, leveraging prior knowledge about the expected behavior of a dynamical system. In this work, we identify and formalize *permissible system states*, a novel type of prior knowledge that is often available for systems in the context of temporal dynamics modeling. This prior knowledge describes dynamic states that the system is allowed to take during its operation. We propose a knowledge-guided multi-state constraint to encode this type of prior knowledge through a loss function, making it applicable to any deep learning model. This approach allows to create an accurate data-driven model with minimal effort and data requirements. We validate the effectiveness of our method by applying it to model the temporal behavior of a gas turbine in response to an input control signal. Our results indicate that the proposed method reduces the prediction error by up to 40%. In addition to reducing the dependency on extensive training data, our method mitigates training randomness and enhances the consistency of predictions with the expected behavior.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; *Knowledge representation and reasoning*; • **Applied computing** → **Engineering**; • **Hardware** → *Energy generation and storage*; • **Mathematics of computing** → *Time series analysis.*

E-Energy '24, June 04–07, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0480-2/24/06
https://doi.org/10.1145/3632775.3661967

KEYWORDS

Dynamical Systems, Dynamics Modeling, Micro Gas Turbine, Domain Knowledge, Physics-Guided Deep Learning

ACM Reference Format:
Pawel Bielski, Aleksandr Eismont, Jakob Bach, Florian Leiser, Dustin Kottonau, and Klemens Böhm. 2024. Knowledge-Guided Learning of Temporal Dynamics and its Application to Gas Turbines. In *The 15th ACM International Conference on Future and Sustainable Energy Systems (E-Energy '24), June 04–07, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3632775.3661967

1 INTRODUCTION

A dynamical system is a system whose state changes over time. Modeling dynamical systems is a fundamental task in science and engineering [4, 14, 29]. Conventionally, these systems are represented using application-specific theory-based models that describe their behavior through mathematical equations. However, when the exact equations describing the system's dynamics cannot be determined, or measuring the necessary system components needed for parameterizing these equations proves impractical or excessively expensive, employing theory-based models becomes infeasible [28, 30].

Deep learning methods are a promising data-driven alternative to theory-based models by learning the dynamical system's behavior directly from data [17, 28, 29]. Their flexibility and ability to learn complex relationships make them highly suitable for modeling various real-world systems [29]. However, despite recent successes, these methods have shortcomings. They can produce physically inconsistent predictions, especially when data availability is limited [4, 8], Additionally, they often struggle to generalize to scenarios not adequately represented in the training data [14, 21].

Very often, one knows more about the expected behavior of the system than just training data [24]. In such cases, knowledge-guided machine learning (KGML) [14, 27–29] can help overcome the limitations of purely data-driven models by systematically incorporating prior knowledge. One widely employed method for incorporating prior knowledge into machine learning is adding knowledge-guided constraints (regularization terms) into the loss function. Recent studies have shown that even simple information integrated in this way can substantially enhance the accuracy, consistency, and generalizability of machine learning models [2, 9, 13, 19, 24].

Despite prior knowledge often being available for dynamical systems, identifying, modeling, and integrating new types of prior knowledge into data-driven modeling is tedious in practice. Difficulties arise from the interdisciplinary nature of the task, which requires extensive collaboration with domain experts. Importantly, there are no guarantees that prior knowledge that is relevant and compatible with the specific dynamics-learning task exists, and even if it does, whether it can be formalized. As a result, there are still many types of prior knowledge relevant for dynamics-learning tasks that have not been explored in the existing literature yet [28].

In this work, we identify and formalize a novel type of prior knowledge relevant for data-driven temporal dynamics modeling of systems encountered in engineering, namely *permissible system states*. This type of prior knowledge characterizes the dynamic states a system is allowed to take during its operation. Deviations from these states are penalized while training the machine learning model, preventing it from adopting behavior inconsistent with the domain. For example, *permissible system states* can be used to inform the model about the expected change rates of an output signal. Such knowledge is particularly useful for modeling the system's behavior in response to an input control signal during transition phases, where the output signal shows a delayed response to a rapid change of an input control signal. This is a common scenario when operating energy systems that are vital for the carbon-free modern energy grid, such as gas turbines, energy storage systems, or power-to-X systems.

Because existing knowledge-guided methods do not allow to model *permissible system states*, we propose a novel *multi-state constraint* to represent and incorporate this type of prior knowledge into a neural network, using a knowledge-guided loss function. Importantly, this proposed method can be applied on top of any deep learning model and can be used with both exact and approximate prior knowledge. Approximate prior knowledge is particularly relevant in practical applications where precisely defining *permissible system states* can be challenging due to measurement uncertainties or the stochastic nature of the dynamical system under consideration. To the best of our knowledge, our work is the first attempt to leverage this type of prior knowledge, both in the context of machine learning and temporal dynamics modeling.

To study the effectiveness of our method, we apply it to model the temporal dynamics of a gas turbine in a scenario where measurements from the internal system components are unavailable, which prevents applying conventional theory-based models. In this context, the prior knowledge is specified as the expected change rate of the gas turbine's output during transition and stationary phases. Our findings indicate that our method allows to create an accurate data-driven model with minimal effort and data requirements. It reduces the prediction error by up to 40%, while simultaneously reducing the need for extensive data and mitigating training randomness, compared to the knowledge-uninformed deep learning baseline. Additionally, the method aids the model to behave consistent with the domain and enhances its ability to generalize to data with varying characteristics, especially in scenarios with limited available training data. We have made the code[1] and the experimental data[2] available online. Additionally, we have contributed the data to the UCI Machine Learning Repository under the name *Micro Gas Turbine Electrical Energy Prediction*. To sum up, our contributions are as follows:

(1) We identify and formalize *permissible system states*, a novel type of prior knowledge that is relevant for data-driven temporal dynamics modeling of engineering systems.
(2) We introduce a novel *multi-state constraint* designed to incorporate this prior knowledge into any neural network, using a knowledge-guided loss function.
(3) We thoroughly evaluate our approach on the task of data-driven temporal dynamics modeling of a gas turbine. This includes performance tests with varied dataset sizes, a sensitivity analysis of hyperparameters, assessing robustness against approximate and incorrect prior knowledge, and a qualitative prediction analysis.

Paper outline. Section 2 features fundamentals. Section 3 describes our approach. Section 4 reports on experiments. Section 5 reviews related work and Section 6 concludes.

2 PRELIMINARIES

In this section, we present fundamentals for our work. First, we introduce the scenarios of temporal dynamics modeling in general (Section 2.1) and gas turbine modeling in particular (Section 2.2). Next, we discuss the limitations of classical deep learning in such scenarios (Section 2.3). Further, we describe the prior knowledge identified in our scenario (Section 2.4). Finally, we explain how loss functions are used in knowledge-guided machine learning (Section 2.5).

2.1 Problem: Temporal Dynamics Modeling

In this study, we focus on discrete-time nonlinear dynamical systems that evolve over time in response to an input control signal. Such a system is characterized by an unknown function f:

$$y_t = f(x_t, x_{t-1}, \ldots, x_{t-N+1}) \tag{1}$$

Here, $x_t \in X$ is the input control signal at time t, and $y_t \in Y$ is the system output signal at time t, which depends on the N last samples of the input control signal. We assume to have access to a dataset comprising tuples of input-output system states $((x_t, y_t) \mid t \in \{1, \ldots, T\})$. T is the number of discrete time samples in the dataset. The objective of temporal dynamics modeling with deep learning is to estimate the unknown function f using a parameterized function f_θ, where $\theta \in \mathbb{R}^p$ are parameters of a neural network, to predict the output y_t given the control input $x_t, x_{t-1}, \ldots, x_{t-N+1}$.

2.2 Our Scenario: Gas Turbine Modeling

Gas turbines play an important role in modern power and heat generation systems, serving various functions such as meeting peak demand, providing backup for renewable energy sources, and enabling decentralized power generation with minimal transportation and conversion losses [3, 10]. Additionally, gas turbines will play a crucial role in future cross-sector power-to-gas energy systems, which enable the conversion of electricity into gas, its storage,

[1] https://github.com/Energy-Theory-Guided-Data-Science/Gas-Turbine

[2] https://doi.org/10.35097/sLJiahifxvfDKMEc

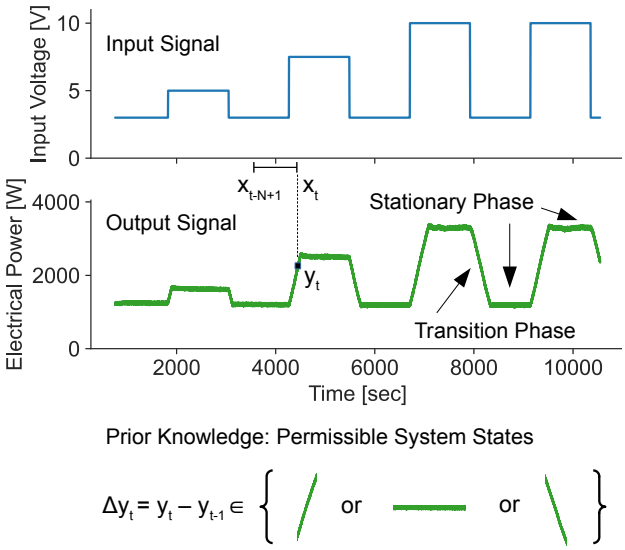


Figure 1: Temporal dynamics modeling of an output signal in response to an input control signal, and the identified prior knowledge in the form of permissible system states.

transportation, and subsequent conversion back to electricity [11]. Their flexibility to operate on a range of fuels, including biofuel and hydrogen, enables them to fulfill these tasks while achieving net-zero carbon emissions [7]. Accurately modeling the temporal behavior of gas turbines enables precise planning of their production capacity, which is a crucial requirement for their effective use in a modern energy grid [10, 20, 22].

In this study, we focus on modeling the behavior of a commercial micro gas turbine designed for residential households, generating approximately 3 kW of electrical power. In our setting, we model the output power in response to an input control signal within the timeframe spanning seconds to hours (see Figure 1). Inputs and outputs are easily obtainable operational data, which eliminates the need for installing measurement sensors within the internal gas turbine components or conducting measurements in a laboratory. Each level of the input control signal corresponds to a stationary, i.e., constant, level of the output power. Notably, a noticeable delay occurs in the output signal during transitions in response to changes in the input control signal.

Conventional theory-based models are not applicable in our scenario, as they require access to the measurements of the internal gas turbine components for parametrization. On the other hand, rule-based and transfer-function methods are susceptible to measurement inaccuracies, or encounter difficulties in accurately representing the system's behavior during transition phases, as discussed in Section 5. Meanwhile, classical data-driven deep learning methods also encounter difficulties, as discussed next.

2.3 Issues with Classical Deep Learning

In real-world situations with limited training data, accurately modeling the temporal behavior of a gas turbine using deep learning methods proves challenging. As depicted in Figure 2, a recurrent

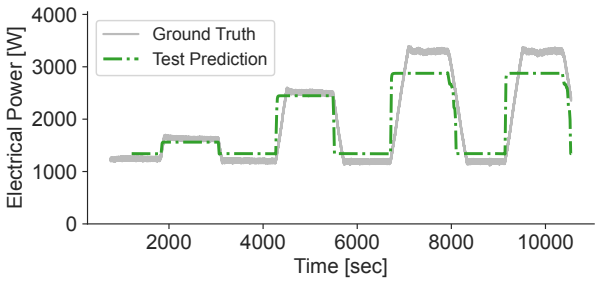


Figure 2: Data-driven methods can struggle to model the transitions when training data is limited.

neural network with LSTM cells, when trained on a small dataset (here: two time series), struggles to accurately replicate the system's behavior during both stationary and transition phases. During stationary phases, the model's predictions deviate from the actual data. The model's predictions during transition phases are inconsistent with prior knowledge, failing to represent transitions as smooth, gradual changes with a constant gradient. This inconsistency with the expected behavior of the gas turbine showcases the shortcomings of knowledge-uninformed deep learning models. In this paper, we illustrate how to improve both the prediction accuracy and consistency on the transitions for deep learning trained on a small dataset, by applying principles from knowledge-guided machine learning.

2.4 Prior Knowledge: Permissible System States

The dynamical state of a system is a characteristic that changes over time. Some dynamical systems have a finite number of dynamical states that the system is allowed to take. We refer to this type of prior knowledge as a finite set of *permissible system states*. Domain expertise about the system can provide insights into whether such *permissible system states* exist, and how to represent and measure them. Depending on factors like measurement uncertainties or the stochastic nature of the system, prior knowledge in that form can be defined either exactly or approximately. The *multi-state constraint* that we introduce in Section 3 informs the machine learning model about these *permissible system states* by penalizing deviations from them, preventing the model from adopting a behavior inconsistent with the domain.

In temporal dynamics modeling, certain systems can be characterized by a finite set of permissible values for their output change rates. For example, in our gas turbine study, the finite set of *permissible system states* consists of three distinct output change rates, as depicted in the lower part of Figure 1. We expect both the rising and falling transition phases to have a constant non-zero change rate, and stationary phases to have a change rate of zero. The values of the change rate for the transition phases can be obtained from ground-truth data by estimating the derivative of the output signal over time. This type of prior knowledge is not unique to gas turbines; similar knowledge is often available for other dynamical systems in mechanical, thermal, or electrical engineering, such as thermal engines, energy storage systems, or power-to-X systems.

2.5 Using Loss Functions in Knowledge-Guided Machine Learning

One of the most widely used paradigms to incorporate prior knowledge into neural networks is via regularization, where the loss function is augmented with additional terms based on prior knowledge [16, 28]:

$$Loss = Loss_{ML} + \lambda_K \cdot Loss_K \tag{2}$$

$Loss_{ML}$ represents a conventional loss function, such as the mean squared error (MSE):

$$Loss_{ML} = \frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2 \tag{3}$$

Such a loss function is commonly employed in regression tasks, i.e., prediction tasks with continuous target variables. It does not consider prior knowledge.

$Loss_K$ has to be designed to guide the neural network towards solutions aligning with domain knowledge or general properties of the target function. The hyperparameter $\lambda_K \in \mathbb{R}$ determines its relative importance compared to $Loss_{ML}$. Typically, λ_K is determined through hyperparameter tuning. The loss term $Loss_K$ is often considered a 'soft' constraint, meaning that it does not have to be strictly met but instead imposes a penalty that increases as the constraint is violated more. We now present two examples of knowledge-guided loss constraints from the literature that encode general properties of the target function.

2.5.1 Example: Approximation constraint. The approximation constraint, as introduced by [2], specifies an expected prediction range between values $a \in \mathbb{R}$ and $b \in \mathbb{R}$. The loss term penalizes predictions outside this range based on their deviation from the range's limits:

$$Loss_{approx} = \frac{1}{T}\sum_{t=1}^{T}\begin{cases}0 & , \ \hat{y}_t \in [a,b] \\ (\hat{y}_t - a)^2 & , \ \hat{y}_t < a \\ (\hat{y}_t - b)^2 & , \ \hat{y}_t > b\end{cases} \tag{4}$$

2.5.2 Example: Monotonicity constraint. The monotonicity constraint, as introduced by [2], defines a monotonic relationship between input and output variables. The loss term penalizes predictions that violate monotonicity, i.e., outputs $\hat{y}$ and $\hat{y}'$ with $\hat{y} > \hat{y}'$ for model inputs x and x' with $x < x'$. In our scenario, we are interested in the output signal over time, i.e., we would set $x := t-1$ and $x' := t$ in the context of the monotonicity constraint. Thus, the loss term would be defined as:

$$Loss_{mono} = \frac{1}{T\text{-}1}\sum_{t=2}^{T}\begin{cases}0 & , \ \hat{y}_{t-1} \le \hat{y}_t \\ (\hat{y}_t - \hat{y}_{t-1})^2 & , \ \hat{y}_{t-1} > \hat{y}_t\end{cases} \tag{5}$$

2.5.3 Inapplicability of the existing constraint types to our scenario. Both discussed constraint types have limitations that make them inapplicable to our scenario in their current form. The approximation constraint does not allow to encode multiple states, as in our case, but only a single range of values. However, its capability to penalize values outside a defined range becomes useful when considering approximate state definitions of the *permissible system states*. On the other hand, the monotonicity constraint is not suitable for non-monotonous functions, as in our case. However, it can penalize inconsistencies between different points of the function, which becomes useful when we define the multi-state constraint on consecutive values.

3 PROPOSED METHOD

In this section, we present our approach. First, we propose general formulations of exact and approximate multi-state constraints (Section 3.1). Second, we show how to adapt these constraints to our scenario, i.e., gas turbine modeling (Section 3.2).

3.1 General Formulation

We define a model state $\tilde{s}$ as a value or vector describing any characteristic of a machine learning model, such as its output. In the context of neural networks, the model state may be the value of a particular neuron, the network's output at a specific layer, or more generally, any combination of these values. We assume we can compute a real-valued distance $d(\cdot)$ between these model states, using a metric like the Euclidean distance. We denote the finite set of permissible model states as $S = \{s^1, s^2, \ldots, s^n\}$. These states represent the prior knowledge we would like the model to adhere to. To inform the neural network about S, we introduce a multi-state constraint as a term in the loss function. We propose two variants: an exact (Section 3.1.1) and an approximate (Section 3.1.2) version.

3.1.1 Exact multi-state constraint. In the exact variant of the constraint (see Equation 6), each permissible state corresponds to a single value. The loss term penalizes model states $\tilde{s}_t$ that do not belong to the set of permissible model states S, imposing a penalty proportional to the deviation from the closest permissible state. If $\tilde{s}_t$ matches a permissible state exactly, then the penalty is zero. The ratio between the hyperparameters β^i of different permissible states allows to influence which of the permissible states is used for penalizing model states in the loss function. We provide more explanations of these hyperparameters in Section 4.6.

$$Loss_{MS} = \frac{1}{T}\sum_{t=1}^{T}\min_{i\in\{1,\ldots,n\}}\beta^i \cdot d(\tilde{s}_t, s^i) \tag{6}$$

3.1.2 Approximate multi-state constraint. For scenarios where the model states S can only be defined approximately, we introduce an approximate multi-state constraint (see Equation 7). Drawing inspiration from the approximation constraint in literature (see Equation 4), we define an approximate state as a range of values, i.e., $s^i = [s^i_{min}, s^i_{max}]$. If a model state $\tilde{s}_t$ lies within the range of any permissible model state, then the penalty is zero. Otherwise, we apply a penalty proportional to the deviation from the nearest state range:

$$Loss_{app-MS} = \frac{1}{T}\sum_{t=1}^{T}\min_{i\in\{1,\ldots,n\}}\beta^i \cdot d_{range}(\tilde{s}_t, s^i)$$
$$\text{where } d_{range}(\tilde{s}_t, s^i) = \begin{cases}0 & , \ s^i_{min} \le \tilde{s}_t \le s^i_{max} \\ d(\tilde{s}_t, s^i_{min}) & , \ \tilde{s}_t < s^i_{min} \\ d(\tilde{s}_t, s^i_{max}) & , \ \tilde{s}_t > s^i_{max}\end{cases} \tag{7}$$

Exact states are a special case of approximate states where the range is zero, i.e., the lower bound also is the upper bound.

3.2 Application to Gas Turbines

In general, it makes sense to define the model state $\tilde{s}$ in a form compatible with the form of the available prior knowledge. In our scenario, prior knowledge takes the form of a finite set of permissible values for the output change rates, as described in Section 2.4 and depicted in the lower part of Figure 1. To make the model state compatible with the notion of change rates, we define the model state $\tilde{s}$ as the difference between two consecutive predictions:

$$\tilde{s}_t = \Delta\hat{y}_t = \hat{y}_t - \hat{y}_{t-1} \tag{8}$$

Consequently, the set of *permissible model states* defines a set of permissible differences $S = \{\Delta^1, \Delta^2, \ldots, \Delta^n\}$ that represent the set of change rates for two consecutive predictions of the model.

For our dataset, we define the set of permissible differences by estimating the derivative of the output signal over time $\Delta y/\Delta t$ for the transitions and assuming a difference of zero for the stationary phases. As a result, we have a set with three permissible differences: $\Delta^+ = 6.388\,\mathrm{W\,s^{-1}}$ for rising transitions, $\Delta^- = -6.388\,\mathrm{W\,s^{-1}}$ for falling transitions, and $\Delta^0 = 0\,\mathrm{W\,s^{-1}}$ for stationary phases (see the lower part of Figure 1). We encode these three permissible system states into the loss function with the *multi-state constraint* (see Equation 6), employing a squared distance to penalize deviations of the model's output from these states:

$$Loss_{MS} = \frac{1}{T}\sum_{t=2}^{T} \min\left(\beta^+(\Delta\hat{y}_t - \Delta^+)^2, \beta^-(\Delta\hat{y}_t - \Delta^-)^2, \beta^0(\Delta\hat{y}_t - \Delta^0)^2\right) \tag{9}$$

This equation can be simplified for our data with $\Delta^+ = -(\Delta^-) = 6.388$, $\Delta^0 = 0$, $\beta^+ = \beta^- = \beta^{trans}$, and $\beta^0 = \beta^{stat}$:

$$Loss_{MS} = \frac{1}{T}\sum_{t=2}^{T} \min\left(\beta^{trans}(|\Delta\hat{y}_t| - 6.388)^2, \beta^{stat}(\Delta\hat{y}_t)^2\right) \tag{10}$$

The previous two equations assume exact state definitions; in our experiments, we also evaluate the approximate variant of the constraint type, defining intervals around the exact values.

4 EXPERIMENTS

In this section, we assess the effectiveness of our method in modeling the temporal dynamics of a gas turbine. First, we describe our experimental design (Section 4.1). Subsequently, we demonstrate that incorporating the multi-state constraint (both its exact and approximate formulations) consistently diminishes the prediction error across varying training-set sizes (Section 4.2). Then, we evaluate the relationship between approximate prior knowledge and prediction error (Section 4.3). Next, we evaluate the method's resilience to incorrect prior knowledge (Section 4.4). We also showcase that our method effectively models transitions in line with prior knowledge, aiding the model in learning patterns that generalize well to test data with different characteristics (Section 4.5). Finally, we conduct a sensitivity analysis of the hyperparameters of our method (Section 4.6).

4.1 Experimental Design

4.1.1 Data. We collected our dataset from a small commercial gas turbine, described in detail in [18]. The dataset comprises time-series data for both the input control signal and the corresponding output (electrical power). Collecting data from gas turbines is a challenging and time-consuming process, requiring the installation of measurement devices and direct supervision by a domain expert for extended periods during experiments. Moreover, real-life operational restrictions on the gas turbine limit the execution of certain test profiles. Our dataset comprises eight time series that depict the gas turbine's behavior under diverse conditions (see Figure 10). The time series vary in duration from 6,495 to 11,820 data points with a resolution of approximately 1 second, corresponding to approximately 1.8 h to 3.3 h. Four *rectangular* time series represent scenarios in which the input control signal changes instantaneously but the output power follows with a visible delay. Accurate modeling of both transitions and stationary phases is crucial for precise gas turbine modeling in these scenarios. The four remaining *continuous* time series represent scenarios in which the input control signal changes gradually and there are no visible delays in the output power. For these time series, learning transitions is less important for modeling the overall behavior.

4.1.2 Prediction model. As our baseline method without prior knowledge, we employ a recurrent neural network (*RNN*) with three hidden layers, each containing 32 LSTM units. The input size is $N = 451$, corresponding to the longest transition length in the dataset. In preliminary experiments, we determined that this architecture has sufficient capacity to learn transitions when trained with six time series, which represents the maximum training-set size in our experiments. Architectures with fewer layers were incapable of modeling transitions, while architectures with more layers did not improve performance but required more training time.

Based on our sensitivity analysis (see Section 4.6), we set the hyperparameter λ_K to 1000 and the weighting of states between transitions and stationary phases to a ratio of $\beta^{stat}/\beta^{trans} = 0.6$ in the other experiments. We trained the models with a learning rate of 0.001 for 300 epochs, using early stopping with a patience of 90. To evaluate prediction error in our experiments, we use the root mean squared error (RMSE).

We implemented the proposed method using *TensorFlow* [1]. On our server with an NVIDIA GeForce GTX 1080 Ti (11 GB Memory), the training took between 11 min (training size of 1) and 90 min (training size of 6), and the inference on one time series took 4 s.

4.2 Impact of Training-Set Size

4.2.1 Experimental design. In this section, we assess prediction error in terms of RMSE across varying training-set sizes. We select two time series (#4, which is rectangular, and #22, which is continuous; see Figure 10) as the test set. From the remaining six time series, we randomly choose one to six time series for training and repeat each training five times to account for randomness in model initialization. We compare the baseline *RNN* with three knowledge-guided variants: (1) RNN^{MS} uses an exact multi-state constraint, representing a scenario with accurate prior knowledge. (2) $RNN^{app-MS,1\%}$ uses an approximate multi-state constraint and an error tolerance of 1%, representing a situation with sufficiently

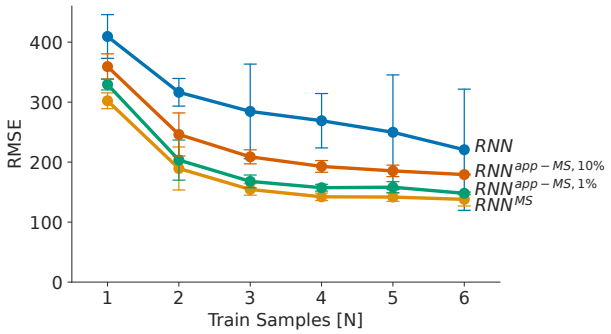


Figure 3: Prediction error (mean RMSE ± standard deviation) over training-set size.

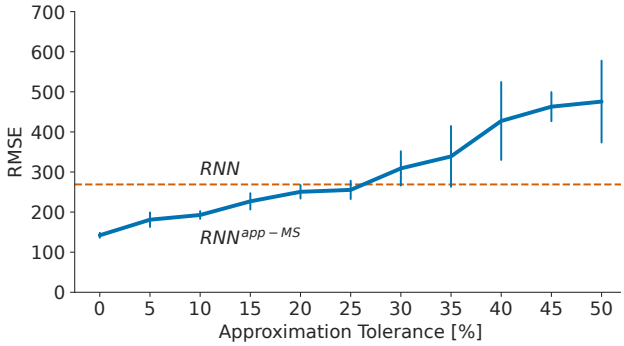


Figure 4: Prediction error (mean RMSE ± standard deviation) over approximation tolerance of prior knowledge.

accurate but not perfect prior knowledge. (3) $RNN^{app-MS,10\%}$ uses an approximate multi-state constraint and an error tolerance of 10%, representing a scenario with higher uncertainty, which also is realistic in practical applications. The value ranges for the approximate constraint are symmetric around the expected value and are defined relative to the difference between the states, which in our scenario is equal to $\Delta^+ - \Delta^0 = \Delta^0 - \Delta^- = 6.388$ For instance, in transition phases of $RNN^{app-MS,10\%}$, with an expected value of $6.388\,\mathrm{W\,s^{-1}}$ and a 10% tolerance, the approximate state is $6.388 \pm 0.6388 = [5.7492, 7.0268]$. For stationary phases, with an expected value of 0 and a 10% tolerance, the approximate state is $0 \pm 0.6388 = [-0.6388, 0.6388]$. The tolerance for $RNN^{app-MS,1\%}$ is calculated similarly.

4.2.2 Results. As depicted in Figure 3, the test prediction error consistently improves with increasing training-set sizes for all three variants of multi-state constraints. In particular, RNN^{MS} and $RNN^{app-MS,1\%}$ reduce the test prediction error by 30-40%, while $RNN^{app-MS,10\%}$ reduces it by 15-20%, compared with RNN, the baseline without prior knowledge. Further, the multi-state constraint methods cause the model to reach a plateau in error with only three to four training samples, whereas the baseline does not plateau even until six samples. Additionally, the multi-state constraint methods RNN^{MS} and $RNN^{app-MS,1\%}$ achieve better test prediction error when trained on two samples than the baseline trained on six samples. This result suggests that integrating prior knowledge effectively reduces the model's dependence on extensive training data and enables reaching the optimum error with less data. Additionally, all multi-state constraint methods show lower variance between different evaluation runs than the baseline, indicating that prior knowledge may help mitigate training randomness. As the amount of training data increases, the performance gap between the baseline and the knowledge-guided approach diminishes, emphasizing the particular benefits of knowledge-guided approaches when data is limited. Furthermore, while both approximate multi-state constraints perform worse than the exact constraint, they still outperform the baseline. These findings confirm that even an approximate definition of prior knowledge can be useful.

4.3 Impact of the Approximation Range

4.3.1 Experimental design. In this section, we assess the impact of the tolerance range of the approximate multi-state constraint on the prediction error. In particular, we vary the width of the range. As in Section 4.2, we use ranges that are symmetric around the expected value and define the tolerance relative to the distance between the states, i.e., $6.388\,\mathrm{W\,s^{-1}}$. We use the same two time series as in Section 4.2 as test set and perform five training repetitions, each time using a random selection of four out of the remaining six time series for each value of the prior knowledge. We evaluate tolerance values from 0% to 50% with 5% increments to explore the method's limits. The tolerance of 0% represents the exact state definition.

4.3.2 Results. As illustrated in Figure 4, the approximate multi-state constraint outperforms the baseline without prior knowledge (depicted as a striped line) in terms of test prediction error even with an approximation range as large as 25%. However, for a tolerance of 50%, where the lower range of one state overlaps with the upper range state of another state, the method performs significantly worse than the baseline. Additionally, we observe that models trained with prior knowledge with smaller tolerance have lower variance in prediction error between different evaluation runs than when using prior knowledge with larger tolerance.

4.4 Impact of Incorrect Prior Knowledge

4.4.1 Experimental design. In this section, we assess the robustness of the exact multi-state constraint when faced with incorrect prior knowledge about the permissible states. This scenario simulates a situation where the provided prior knowledge is exact but does not match the reality. We use the exact multi-state constraint and vary the values for the prior knowledge, i.e., the change rates of the gas turbine's output signal at the transitions. Again, we use the same two time series as in Section 4.2 as test set and perform five training repetitions, each time using a random selection of four out of the remaining six time series for each value of the prior knowledge. We test the correct value of $\Delta^+ = 6.388\,\mathrm{W\,s^{-1}}$, as well as exponentially scaled values up to 128 times smaller or larger.

4.4.2 Results. As depicted in Figure 5, the exact multi-state constraint outperforms the baseline without prior knowledge (depicted

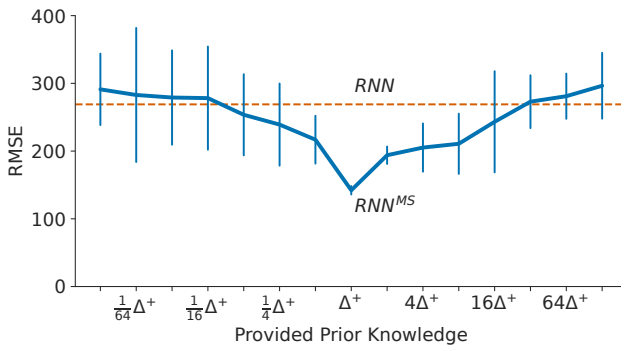


Figure 5: Prediction error (mean RMSE ± standard deviation) over incorrect prior knowledge (correct prior knowledge in center).

as a striped line) regarding test prediction error even if the assumed change rate is eight times smaller or larger than in reality. However, when the deviation from the correct value becomes larger, the method performs worse than the baseline. These results suggest that even incorrect prior knowledge can be beneficial within a certain range. Additionally, we observe that the model trained with the correct prior knowledge has lower variance between different evaluation runs than with incorrect prior knowledge. Finally, the U-shaped relationship suggests that it could be technically feasible to identify the optimal value of prior knowledge by sampling different values and observing the prediction error. This could, in turn, reduce the method's reliance on providing the exact value of prior knowledge beforehand.

4.5 Qualitative Evaluation of Predictions

4.5.1 Experimental design. In this section, we showcase the effectiveness of our proposed method qualitatively. We demonstrate that our approach not only reduces the prediction error but also addresses the challenges of accurately representing stationary phases and providing consistent predictions for transition phases, as discussed in Section 2.3 and depicted in Figure 2. In particular, we visualize the test predictions of the exact multi-state RNN (RNN^{MS}) and the baseline (RNN) with the lowest prediction error out of the five training repetitions from the experiments in Section 4.2. To show how the impact of our method depends on the training-set size, we visualize predictions for training sizes of two and six. Figure 10 plots the corresponding input signals.

4.5.2 Results. The first row of Figure 6a shows that the baseline model RNN, trained on data from two time series, fails to accurately model the rising transition, representing it as a discrete step rather than a gradual change. We also observe that RNN does not capture the stationary levels accurately. The second row of Figure 6a shows results from training with a multi-state constraint, resulting in a significant RMSE reduction for both rectangular and continuous time series. Further, the representation of transitions is smoother, and the overall behavior of the gas turbine is captured better. In the first row of Figure 6b, we see that while the baseline trained on data from six time series is able to capture transitions of the rectangular test-set time series as a gradual change, it does not capture the stationary levels accurately, which results in a worse RMSE than RNN^{MS} has. For the continuous test-set time series, RNN^{MS} also is slightly better at following the ground truth data.

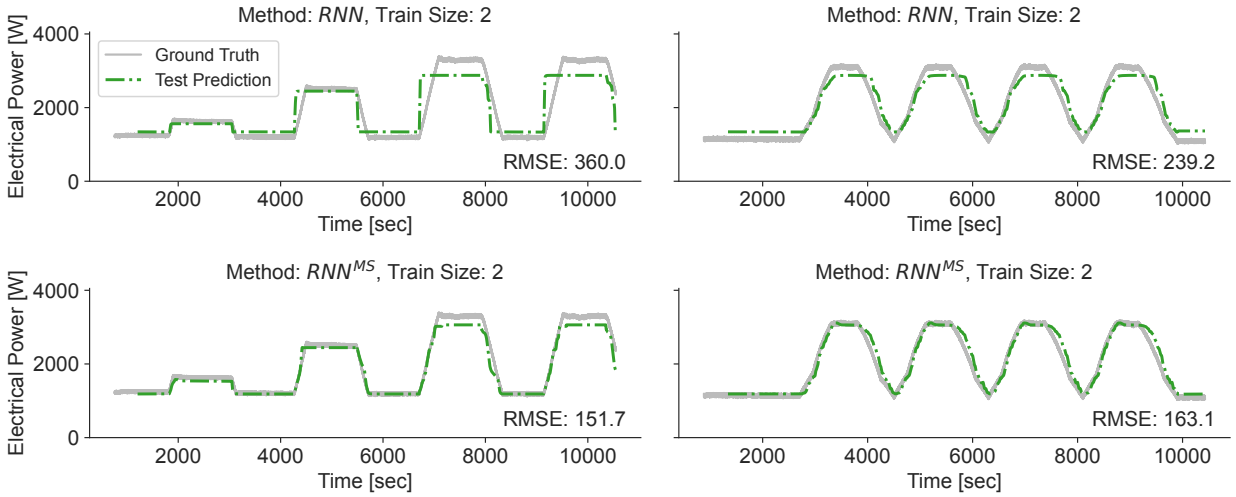
To sum up, our method proves advantageous since it improves the modeling of transitions as well as stationary phases, especially when the training-set size is small. In practical applications, we recommend training the model multiple times and choosing the best one to mitigate the effects of randomness in network initialization.

4.6 Sensitivity Analysis

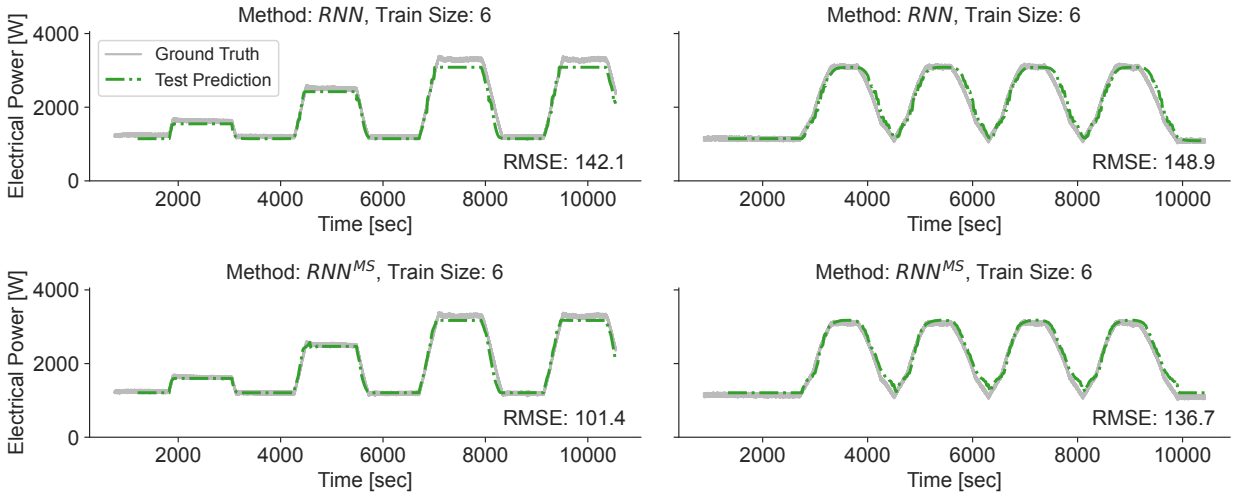
4.6.1 Experimental design. In this section, we investigate the two hyperparameters of our method: λ_K and the ratio $\beta^{stat}/\beta^{trans}$. For both hyperparameters, we evaluate an exponentially scaled range of values, i.e., $\{0.001, 0.01, \ldots, 10000\}$. As discussed in Section 2.5, λ_K determines the relative importance of the knowledge-guided loss $Loss_K$ in relation to $Loss_{ML}$. The ratio $\beta^{stat}/\beta^{trans}$ between the hyperparameter β^i of different permissible states allows to determine which of the permissible states is used to compute the penalty in the loss function, by influencing the decision border between different states (see Section 3.1.1). A ratio of 1 positions the decision boundary precisely halfway between the values for transitions ($\Delta^+ = -\Delta^- = 6.388\,\mathrm{W\,s^{-1}}$) and stationary phases ($\Delta^0 = 0\,\mathrm{W\,s^{-1}}$). Ratios less than one shift the border toward the transition value and ratios greater than one shift it toward the stationary value. The relationship between this ratio, the decision border, and the method's performance can be complex, underscoring the importance of hyperparameter tuning to find optimal values. In our analysis, we use the same two time series as in Section 4.2 as test set and perform five training repetitions, each time using a random selection of four out of the remaining six time series for each hyperparameter value under examination.

4.6.2 Results: λ_K. As shown in Figure 7, the model's prediction error with the multi-state constraint approaches the baseline's error for small values of the parameter λ_K. This behavior is expected, as for small λ_K values, the knowledge-guided loss $Loss_K$ has a small impact on model training. As λ_K approaches 1000, the prediction error decreases, reaching a minimum. However, for $\lambda_K = 10000$, the error increases significantly. This behavior aligns with expectations, as excessively high values of λ_K cause the knowledge-guided loss $Loss_K$ to dominate over the machine learning loss $Loss_{ML}$, leading the model to disregard learning the correct stationary phases (see Figure 9c).

4.6.3 Results: $\beta^{stat}/\beta^{trans}$. In Figure 8, we observe that when the ratio $\beta^{stat}/\beta^{trans}$ is below or equal to 1, the prediction error remains consistently low, with the exact value of $\beta^{stat}/\beta^{trans}$ having minimal impact. However, for values greater than 1, the prediction error increases significantly. The results suggest that in our scenario, optimal results are achieved when the decision border is positioned closer to the value for transitions ($\Delta^+ = -\Delta^- = 6.388\,\mathrm{W\,s^{-1}}$) than the stationary phase ($\Delta^0 = 0\,\mathrm{W\,s^{-1}}$). Placing the decision border too close to the state for the stationary phase may force predictions with output change rates that are slightly above zero to converge to the output change rate for transitions, which negatively affects the predictions for the continuous time series in particular (see



(a) Trained on two time series from our dataset.



(b) Trained on six time series from our dataset.

Figure 6: The predictions of the best of five models.

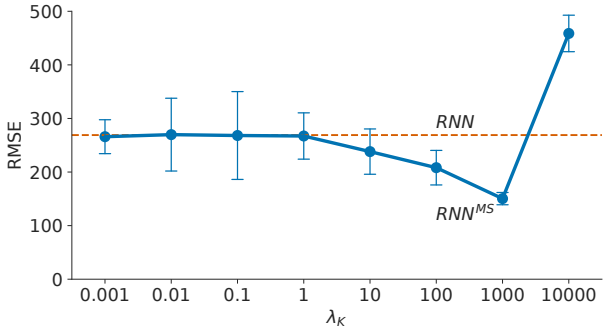


Figure 7: Prediction error over hyperparameter λ_K for $\beta^{stat}/\beta^{trans} = 1$.

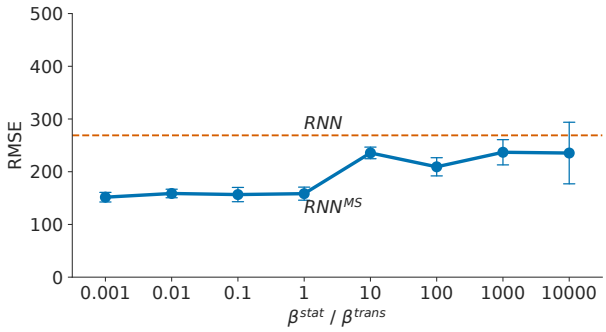


Figure 8: Prediction error over hyperparameter $\beta^{stat}/\beta^{trans}$ for $\lambda_K = 1000$.

Figure 9d). In our scenario, the typical inconsistency in model prediction arises from predicted transitions being too steep (see Figure 2). As a result, placing the decision border too close to the state for the transition phase does not have such a negative impact because predictions converge to the closest permissible state, which is the transition state, regardless of the positioning of the decision border.

5 RELATED WORK

In this section, we discuss related work from two areas: using prior knowledge in loss functions (Section 5.1) and gas turbine modeling (Section 5.2).

5.1 Integrating Prior Knowledge in the Loss Function for Modeling Dynamical Systems

There is a rapid growth of papers on incorporating prior knowledge through physics-guided constraints in the loss function and using them in various scenarios [14, 27–29]. Early work in this area focused on defining generic types of prior knowledge independently from the application domain. For instance, [24] introduced a knowledge-guided loss function to inform a neural network about the derivatives at specific points of a generic target function. Soon after, [2] proposed constraints on the loss function to encode invariances, monotonic relationships, and permissible value ranges. These generic, domain-independent types of prior knowledge enabled modeling application-specific prior knowledge in various data-driven scenarios of modeling particular dynamical systems. For example, [19] employed a combination of monotonicity and approximation constraints for predicting oxygen solubility in water, while [12] applied a monotonicity constraint on the density-depth relationship when predicting temperature in lakes.

Much less research has been done on identifying and formalizing prior knowledge that is applicable across various dynamical systems. For example, [15] proposed a constraint to encode the energy conservation law. In general, proposing and evaluating new types of prior knowledge and formalizing them with the knowledge-guided constraints in the loss function remains an active field of research [28]. Such research is particularly impactful if the proposed prior knowledge is applicable to a broader class of problems, allowing for the reuse of the knowledge-guided methods. To the best of our knowledge, our work is the first to identify and propose the use of prior knowledge in the form of permissible system states. None of the previously proposed knowledge-guided constraints are suitable for representing this specific type of prior knowledge. While we evaluate our method with modeling a gas turbine, the concept of permissible systems states is applicable for data-driven temporal dynamics modeling of various dynamical systems.

There also is related work that focuses on integrating multiple sources of prior knowledge. For example, [2] balanced multiple knowledge-guided constraints by alternating between them during training, while [19] incorporated multiple constraints as a weighted sum. Additionally, [9] introduced a method for optimizing weights among competing knowledge-guided constraints. However, these approaches do not consider incorporating information about alternative characteristics that a target function can possess, like the permissible system states featured in our approach.

5.2 Gas Turbine Modeling

Theory-based models are the conventional way to model the behavior of gas turbines, e.g., Rowen's Model [22], the IEEE Model [6], or the CIGRE Model [5]. These application-specific, equation-based models accurately capture the temporal behavior of gas turbines by modeling the interaction among various system components and operational conditions. This process depends on obtaining measurements from internal system components using physical sensors and involves significant parametrization efforts [23, 26].

As an alternative of theory-based models, some researchers have turned to widely applicable, data-driven deep learning models to directly predict the output power of gas turbines from operational data. While this approach has shown promise in reducing parametrization and calibration efforts compared to theory-based models, previous studies evaluating these methods focused on scenarios where measurements of the internal system components were accessible [20, 25]. However, this access may not always be guaranteed due to the associated high effort and costs, for example in the concept phases of planning decentralized energy systems [10].

Much less research has been done for modeling scenarios without access to internal system component measurements. [18] explored the use of empirical rule-based and transfer-function models for gas turbine modeling using only easily obtainable input-control and output power data. However, the proposed rule-based ramp model is highly susceptible to measurement uncertainties, and the transfer-function model struggles to optimize for both small and large output changes simultaneously, resulting in imprecise predictions on transitions. For this scenario, we propose an alternative data-driven knowledge-guided deep learning approach that, unlike the transfer-function model, models transitions accurately, and offers broader applicability and robustness against measurement uncertainties than rule-based methods. Further, our method integrates prior knowledge to address the limitations of knowledge-uninformed deep learning. This approach allows us to leverage the minimal manual effort associated with data-driven modeling while creating an accurate model with minimal data requirements.

6 CONCLUSIONS

Permissible system states characterize the states a dynamical system is allowed to take during its operation. Existing knowledge-guided machine learning methods do not allow this type of prior knowledge to be modeled. We formalize permissible system states and propose a novel multi-state constraint that incorporates permissible states into neural networks via a knowledge-guided loss function. This approach allows for the creation of a data-driven model that is accurate with minimal effort and data requirements. We evaluate our method by applying it to model the temporal dynamics of a gas turbine in response to an input control signal. Our results indicate that the proposed method reduces the prediction error by up to 40% compared to the knowledge-uninformed deep learning baseline. It also reduces the dependency on extensive training data, mitigates training randomness, and enhances model consistency with prior knowledge. Further, we show that our method can also tolerate approximate or incorrect prior knowledge to a certain extent. As future work, we aim to explore the applicability of our approach

across various energy systems with similar characteristics, such as thermal engines, energy storage systems, and power-to-X systems.

ACKNOWLEDGMENTS

This work was supported by the German Research Foundation (DFG) as part of the Research Training Group GRK 2153: *Energy Status Data – Informatics Methods for its Collection, Analysis and Exploitation.*

REFERENCES

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. http://arxiv.org/abs/1603.04467 arXiv:1603.04467 [cs].

[2] Yaser Abu-Mostafa. 1992. A Method for Learning From Hints. In *Advances in Neural Information Processing Systems*, Vol. 5. Morgan-Kaufmann. https://proceedings.neurips.cc/paper/1992/hash/7750ca3559e5b8e1f44210283368fc16-Abstract.html

[3] Reyhaneh Banihabib and Mohsen Assadi. 2022. The role of micro gas turbines in energy transition. *Energies* 15, 21 (2022), 8084. https://www.mdpi.com/1996-1073/15/21/8084

[4] Tom Beucler, Michael Pritchard, Stephan Rasp, Jordan Ott, Pierre Baldi, and Pierre Gentine. 2021. Enforcing Analytic Constraints in Neural Networks Emulating Physical Systems. *Physical Review Letters* 126, 9 (March 2021), 098302. https://doi.org/10.1103/PhysRevLett.126.098302

[5] CIGRE Task Force C4.02.25. 2003. Modeling of gas turbines and steam turbines in combined cycle power plants. *CIGRE Technical Brochure* 238 (2003).

[6] F. P. De Mello and D. J. Ahner. 1994. Dynamic models for combined cycle plants in power system studies. *IEEE Transactions on Power Systems (Institute of Electrical and Electronics Engineers);(United States)* 9, 3 (1994). https://www.osti.gov/biblio/6912824

[7] Roberta De Robbio. 2023. Micro Gas Turbine Role in Distributed Generation with Renewable Energy Sources. *Energies* 16, 2 (Jan. 2023), 704. https://doi.org/10.3390/en16020704

[8] Franck Djeumou, Cyrus Neary, Eric Goubault, Sylvie Putot, and Ufuk Topcu. 2022. Neural Networks with Physics-Informed Architectures and Constraints for Dynamical Systems Modeling. http://arxiv.org/abs/2109.06407

[9] Mohannad Elhamod, Jie Bu, Christopher Singh, Matthew Redell, Abantika Ghosh, Viktor Podolskiy, Wei-Cheng Lee, and Anuj Karpatne. 2022. *CoPhy* -PGNN: Learning Physics-guided Neural Networks with Competing Loss Functions for Solving Eigenvalue Problems. *ACM Transactions on Intelligent Systems and Technology* 13, 6 (Dec. 2022), 1–23. https://doi.org/10.1145/3530911

[10] Satya Gopisetty and Peter Treffinger. 2016. Generic Combined Heat and Power (CHP) Model for the Concept Phase of Energy Planning Process. *Energies* 10, 1 (Dec. 2016), 11. https://doi.org/10.3390/en10010011

[11] Manuel Götz, Jonathan Lefebvre, Friedemann Mörs, Amy McDaniel Koch, Frank Graf, Siegfried Bajohr, Rainer Reimert, and Thomas Kolb. 2016. Renewable Power-to-Gas: A technological and economic review. *Renewable Energy* 85 (Jan. 2016), 1371–1390. https://doi.org/10.1016/j.renene.2015.07.066

[12] Xiaowei Jia, Jared Willard, Anuj Karpatne, Jordan Read, Jacob Zwart, Michael Steinbach, and Vipin Kumar. 2019. Physics guided RNNs for modeling dynamical systems: A case study in simulating lake temperature profiles. In *Proceedings of the 2019 SIAM International Conference on Data Mining*. SIAM, 558–566. https://doi.org/10.1137/1.9781611975673.63

[13] Xiaowei Jia, Jared Willard, Anuj Karpatne, Jordan S. Read, Jacob A. Zwart, Michael Steinbach, and Vipin Kumar. 2021. Physics-Guided Machine Learning for Scientific Discovery: An Application in Simulating Lake Temperature Profiles. *ACM/IMS Transactions on Data Science* 2, 3 (Aug. 2021), 1–26. https://doi.org/10.1145/3447814

[14] Anuj Karpatne, Gowtham Atluri, James H. Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. 2017. Theory-Guided Data Science: A New Paradigm for Scientific Discovery from Data. *IEEE Transactions on Knowledge and Data Engineering* 29, 10 (Oct. 2017), 2318–2331. https://doi.org/10.1109/TKDE.2017.2720168 Conference Name: IEEE Transactions on Knowledge and Data Engineering.

[15] Anuj Karpatne, William Watkins, Jordan Read, and Vipin Kumar. 2017. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431* 2 (2017). https://arxiv.org/abs/1710.11431

[16] Karthik Kashinath, M. Mustafa, Adrian Albert, J. L. Wu, C. Jiang, Soheil Esmaeilzadeh, Kamyar Azizzadenesheli, R. Wang, A. Chattopadhyay, and A. Singh. 2021. Physics-informed machine learning: case studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A* 379, 2194 (2021), 20200093. https://doi.org/10.1098/rsta.2020.0093

[17] J. Zico Kolter and Gaurav Manek. 2019. Learning Stable Deep Dynamics Models. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/hash/0a4bbceda17a6253386bc9eb45240e25-Abstract.html

[18] Dustin Kottonau. 2023. *Echtzeitsimulation und Netzintegration einer Mikrogasturbine*. PhD Thesis. Karlsruher Institut für Technologie (KIT). https://doi.org/10.5445/IR/1000156019/v2

[19] Nikhil Muralidhar, Mohammad Raihanul Islam, Manish Marwah, Anuj Karpatne, and Naren Ramakrishnan. 2018. Incorporating Prior Domain Knowledge into Deep Neural Networks. In *2018 IEEE International Conference on Big Data (Big Data)*. 36–45. https://doi.org/10.1109/BigData.2018.8621955

[20] Mostafa Pasandideh, Matthew Taylor, Shafiqur Rahman Tito, Martin Atkins, and Mark Apperley. 2024. Predicting Steam Turbine Power Generation: A Comparison of Long Short-Term Memory and Willans Line Model. *Energies* 17, 2 (2024), 352. https://doi.org/10.3390/en17020352

[21] Haakon Robinson, Suraj Pawar, Adil Rasheed, and Omer San. 2022. Physics guided neural networks for modelling of non-linear dynamics. *Neural Networks* 154 (2022), 333–345. https://doi.org/10.1016/j.neunet.2022.07.023

[22] W. I. Rowen. 1983. Simplified Mathematical Representations of Heavy-Duty Gas Turbines. *Journal of Engineering for Power* 105, 4 (Oct. 1983), 865–869. https://doi.org/10.1115/1.3227494

[23] Emam Shalan, Mohamed Moustafa Hassan, and ABG Bahgat. 2011. Parameter Estimation and Dynamic Simulation Of Gas Turbine Model In Combined Cycle Power Plants Based On Actual Operational Data. *Journal of American Science* 7 (Jan. 2011). https://www.researchgate.net/publication/239523797_Parameter_Estimation_and_Dynamic_Simulation_Of_Gas_Turbine_Model_In_Combined_Cycle_Power_Plants_Based_On_Actual_Operational_Data

[24] Patrice Simard, Bernard Victorri, Yann LeCun, and John Denker. 1991. Tangent Prop - A formalism for specifying selected invariances in an adaptive network. In *Advances in Neural Information Processing Systems*, Vol. 4. Morgan-Kaufmann. https://proceedings.neurips.cc/paper/1991/hash/65658fde58ab3c2b6e5132a39fae7cb9-Abstract.html

[25] Lei Sun, Tianyuan Liu, Yonghui Xie, Di Zhang, and Xinlei Xia. 2021. Real-time power prediction approach for turbine using deep learning techniques. *Energy* 233 (Oct. 2021), 121130. https://doi.org/10.1016/j.energy.2021.121130

[26] M.R.B. Tavakoli, B. Vahidi, and W. Gawlik. 2009. An Educational Guide to Extract the Parameters of Heavy Duty Gas Turbines Model in Dynamic Studies Based on Operational Data. *IEEE Transactions on Power Systems* 24, 3 (Aug. 2009), 1366–1374. https://doi.org/10.1109/TPWRS.2009.2021231

[27] Laura von Rueden, Sebastian Mayer, Katharina Beckh, Bogdan Georgiev, Sven Giesselbach, Raoul Heese, Birgit Kirsch, Julius Pfrommer, Annika Pick, Rajkumar Ramamurthy, Michal Walczak, Jochen Garcke, Christian Bauckhage, and Jannis Schuecker. 2023. Informed Machine Learning – A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems. *IEEE Transactions on Knowledge and Data Engineering* 35, 1 (Jan. 2023), 614–633. https://doi.org/10.1109/TKDE.2021.3079836 Conference Name: IEEE Transactions on Knowledge and Data Engineering.

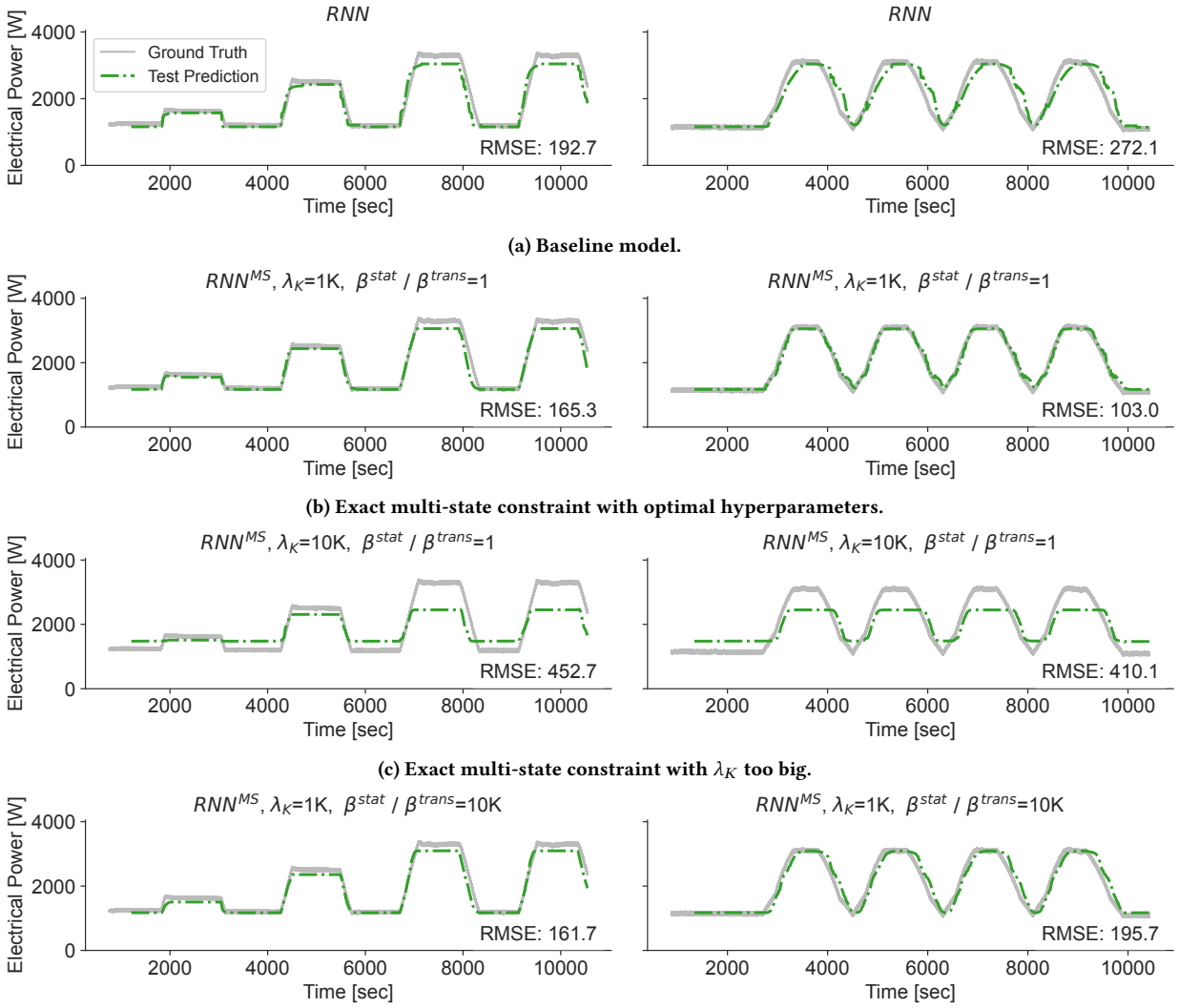
[28] Rui Wang and Rose Yu. 2022. Physics-Guided Deep Learning for Dynamical Systems: A Survey. http://arxiv.org/abs/2107.01272

[29] Jared Willard, Xiaowei Jia, Shaoming Xu, Michael Steinbach, and Vipin Kumar. 2023. Integrating Scientific Knowledge with Machine Learning for Engineering and Environmental Systems. *Comput. Surveys* 55, 4 (April 2023), 1–37. https://doi.org/10.1145/3514228

[30] Yuan Yin, Vincent Le Guen, Jérémie Dona, Emmanuel de Bézenac, Ibrahim Ayed, Nicolas Thome, and Patrick Gallinari. 2021. Augmenting physical models with deep networks for complex dynamics forecasting*. *Journal of Statistical Mechanics: Theory and Experiment* 2021, 12 (Dec. 2021), 124012. https://doi.org/10.1088/1742-5468/ac3ae5

A APPENDIX: ADDITIONAL FIGURES

Received 2 February 2024; revised 3 April 2024; accepted 16 April 2024



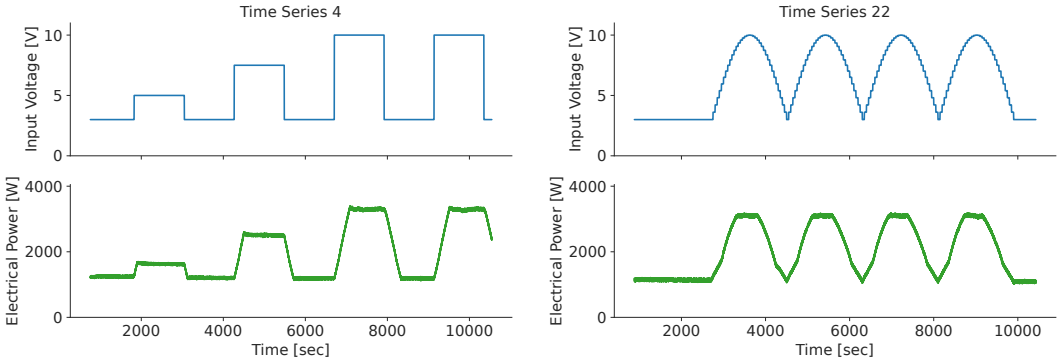
(a) Baseline model.

(b) Exact multi-state constraint with optimal hyperparameters.

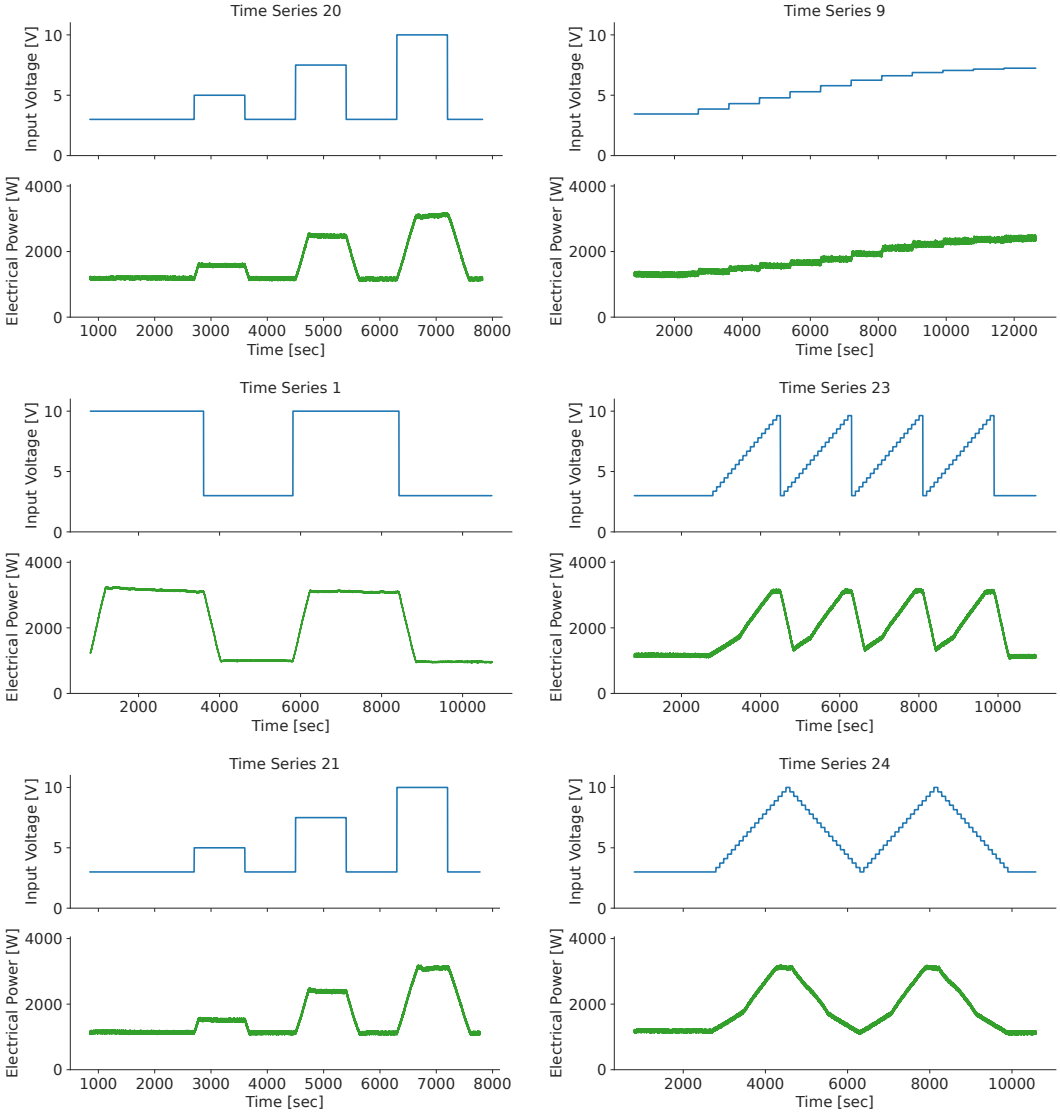
(c) Exact multi-state constraint with λ_K too big.

(d) Exact multi-state constraint with β^{stat} / β^{trans} too big.

Figure 9: The predictions of the best of five models trained on four time series.



(a) Time-series data used for testing.



(b) Time-series data used for training.

Figure 10: All available time series in our dataset.