

CNN-based Compressor Mass Flow Estimator in Industrial Aircraft Vapor Cycle System

Justin Reverdi^{1,2,3}, Sixin Zhang¹, Saïd Aoues², Fabrice Gamboa³, Serge Gratton¹, and Thomas Pellegrini¹

¹Institut de Recherche en Informatique de Toulouse (IRIT)

²LIEBHERR Aerospace Toulouse

³Institut de Mathématiques de Toulouse (IMT)

Abstract—In Vapor Cycle Systems, the mass flow sensor plays a key role for different monitoring and control purposes. However, physical sensors can be inaccurate, heavy, cumbersome, expensive or highly sensitive to vibrations, which is especially problematic when embedded into an aircraft. The conception of a virtual sensor, based on other standard sensors, is a good alternative. This paper has two main objectives. Firstly, a data-driven model using a Convolutional Neural Network is proposed to estimate the mass flow of the compressor. We show that it significantly outperforms the standard Polynomial Regression model (thermodynamic maps), in terms of the standard MSE metric and Engineer Performance metrics. Secondly, a semi-automatic segmentation method is proposed to compute the Engineer Performance metrics for real datasets, as the standard MSE metric may pose risks in analyzing the dynamic behavior of Vapor Cycle Systems.

Keywords—Deep Learning, Convolutional Neural Network, Vapor Cycle System, Virtual Sensor

NOMENCLATURE

\dot{m}	Compressor Mass Flow	kg s^{-1}
ω	Compressor Motor Speed	rad s^{-1}
C	Compressor Motor Torque	N m
P_{in}	Compressor Inlet Pressure	Pa
P_{out}	Compressor Outlet Pressure	Pa
T_{in}	Compressor Inlet Temperature	K
T_{out}	Compressor Outlet Temperature	K

I. INTRODUCTION

IN physical systems, it is crucial to measure certain quantities. Traditionally, physical sensors have been used for this purpose, but they can be expensive, cumbersome, and require maintenance. Sometimes there are no physical sensors for the quantity of interest. As an alternative, virtual sensors have emerged as a cost-effective and reliable method for measuring physical data based on other physical sensors [1]. In complex physical systems, it is sometimes hard to derive physically meaningful equations to construct a virtual sensor. Therefore, using machine learning models to design a virtual sensor has become a popular research direction in recent years [1]–[7]. In this paper, we propose and evaluate a virtual sensor based on Machine Learning (ML) for estimating an important physical quantity (Compressor Mass Flow) in the Vapor Cycle System (VCS) of LIEBHERR in airplanes.

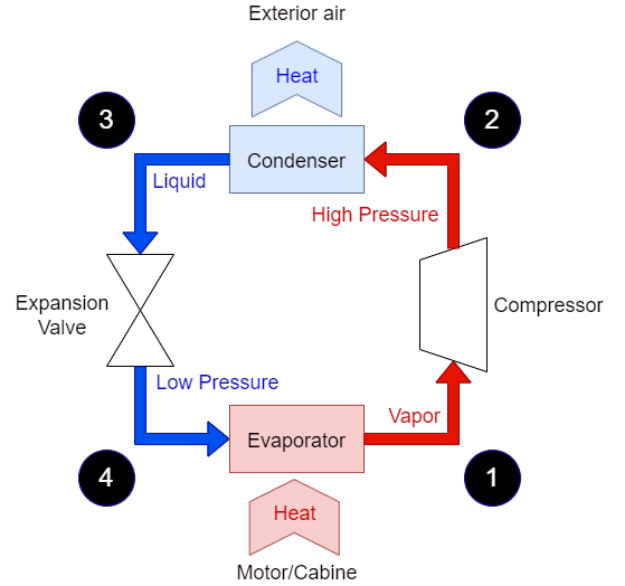


Fig. 1: Vapor Cycle System (VCS)

Background of the VCS: The function of this system is to reduce and control the temperature of a medium by transferring its heat to the refrigerant (Evaporator) and then to transfer the heat from the refrigerant to an external medium (Condenser). The Compressor and the Expansion Valve bring the refrigerant fluid in the appropriate conditions of pressure and temperature for heat transfer. A typical refrigeration cycle comprises four stages through which the refrigerant undergoes: compression (from point 1 to point 2), condensation (from point 2 to point 3), expansion (from point 3 to point 4), and evaporation (from point 4 back to point 1) [8], [9]. This system is represented in Figure 1. The cycle is defined by these transitions that are represented in Figure 2. VCSs are used in aircraft for cooling the cabin and the motors down. For control purposes, it is necessary to accurately measure the mass flow through the compressor with a small response time, lest it could lead to control instability [10] and surge phenomenon [11]. This mass flow can be expressed as the ratio of the power supplied to the fluid by the enthalpy difference between the outlet and the

inlet of the compressor [2] :

$$\dot{m} = \frac{C\omega(1 - \alpha_{loss})}{h_2 - h_1} = \frac{C\omega(1 - \alpha_{loss})}{h(T_{out}, P_{out}) - h(T_{in}, P_{in})} \quad (\star)$$

where $h(P, T)$ is the enthalpy of the refrigerant fluid in vapor phase at temperature T and pressure P . α_{loss} is the proportion of dissipated energy and cannot be measured.

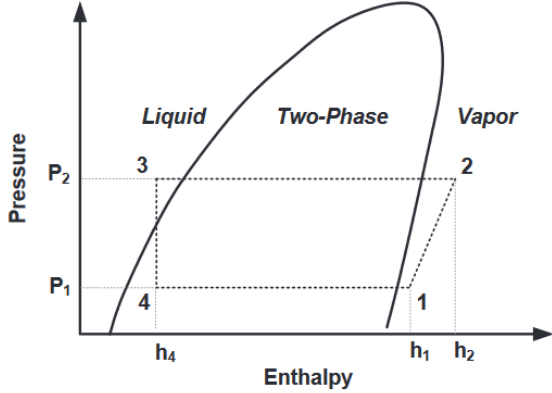


Fig. 2: Vapor Cycle on Pressure-Enthalpy diagram [9]

On the ML-based virtual sensor: There are two standard physical mass flow sensors, the Coriolis [12] and Venturi [13] flowmeter. The first one is highly accurate but not robust to vibrations and hence not adapted to aircraft systems. The second one requires adding a bottleneck that induces a loss of charge and deteriorates the efficiency of the system. This leads to the need for a virtual sensor based on other embeddable sensors. Unfortunately, there are no obvious physical equations that give the flow in function of these measured quantities with sufficient accuracy.

In the laboratory of LIEBHERR Aerospace in Toulouse, experiments were done on an industrial VCS. Several physical quantities were recorded at a high frequency during 30 hours, including the mass flow measured with a Coriolis flowmeter. Indeed, even though the Coriolis flowmeter is not embeddable in aircraft, it is an excellent sensor for measuring the mass flow during the experiment in the laboratory. Those measures pave the way to adopt a supervised machine learning framework to predict the mass flow from other physical quantities (defined in the Nomenclature and exposed in Figure 3). Similar experiments were done in [2] without the torque measure C and in [3] with additional data on the Condenser and the Expansion Valve.

State-of-the-art and main contributions: In [2] an evaluation of statistical models based on Polynomial Regression (PR) is conducted. A CNN based on 2d convolutional networks (usually used to analyze images) is tested in [3] and it is shown to outperform the PR models. The issue with the existing CNN approach is that no guarantee of physical properties are analyzed. In this article, we propose a different CNN architecture to guarantee time-translational invariance, insensitive to the order of input features, and causal properties for real-time estimation. The architecture is adapted from [6]

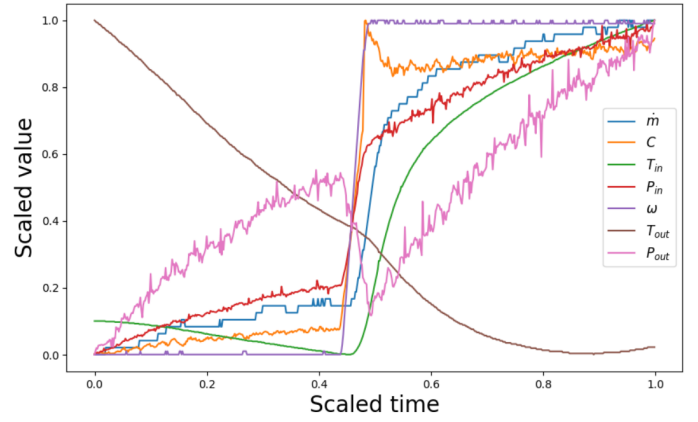


Fig. 3: Scaled measures

for unsupervised learning of time series, with applications to classification and regression problems.

By expanding the scope of applications, one finds in the literature several models for designing virtual sensors [4]–[7], [14]. In [5], a CNN with recurrent skip connections is trained on simulated data, fine-tuned on real data to estimate physical quantities in electrical induction motors, and evaluated with electrical engineering metrics.

The main contribution of this article is to show that the proposed CNN significantly outperforms the PR model on the novel LIEBHERR dataset. To compare the difference between these models, we further propose a semi-automatic evaluation method to compute similar engineering metrics as in [5]. The novelty is that we work on a real dataset so it is needed to introduce a segmentation step of time series based on peak detection.

The rest of the article is organized as follows. In Section II, PR and the proposed CNN models are detailed. In Section III, a two-fold evaluation method is explained, integrating machine learning metrics and engineering ones. The engineer's metrics are computed after a segmentation step, that enables to compute a multitude of metrics and to perform statistics analysis. Finally, the results are presented and discussed in Section IV.

II. ML MODELS

We aim to develop a model capable of predicting real-time mass flow \dot{m} based on specific measurements. Although equation (\star) relates these variables, the quantity α_{loss} is absent from our measurements. To circumvent this lack of physical relationship, we will rely on experimental data to construct two statistical models: Polynomial Regression (PR) and Convolutional Neural Networks (CNN). Each recording consists of a variable-length, multivariate time series containing seven measures, as represented in Figure 3.

Polynomial Regression: Statistical models are used in thermodynamics to determine the state function of a system at equilibrium from experiments. These models are called maps, and a standard way to create them is Polynomial Regression (PR) [15]. Our baseline approach is PR as in [2], [3], [15]

and hence the model is static. It involves fitting a second-order polynomial function without cross-terms to the data points obtained from experimental recordings. This function was then used to predict the mass flow on the test recordings for evaluation. It is important to note that the prediction at time t depends only on the measures at time t .

$$\hat{m}_t = \alpha + \sum_{i=1}^6 \beta_i x_t^{(i)} + \sum_{i=1}^6 \gamma_i \left(x_t^{(i)}\right)^2,$$

where \hat{m}_t is the estimated mass flow at time t , $x_t = (C_t, \omega_t, P_{in,t}, T_{in,t}, P_{out,t}, T_{out,t})$ is the vector of input measures at time t , $\alpha \in \mathbb{R}$ and $\beta, \gamma \in \mathbb{R}^6$ are the model parameters.

In [2], the authors propose to use the equation (\star) by estimating the missing value α_{loss} with a polynomial of other measures and found more precise and consistent results. Combining physical and statistical approach could be interesting, but this is not in the scope of this paper.

Convolutional Neural Network [16], [17]: A Convolutional Neural Network (CNN) is a series of convolution layers with adjustable coefficients, designed to minimize a loss function during the training process. We shall first specify the convolutional layer that we use in this paper and then analyze its physical properties. We then specify the proposed CNN architecture and discuss its advantage with respect to the standard Polynomial Regression model.

Let $x \in \mathbb{R}^{I \times T}$ with x_t^i the i -th feature at time $t \in \{1, \dots, T\}$. The causal one-dimensional convolution (**Conv1D**) is defined as

$$y_t^j = \text{Conv1D}(x) = \sum_{1 \leq i \leq I} \sum_{s=0}^{k-1} x_{t-s}^i w_s^{j,i},$$

where y_t^j is the j -th output channel at time t , $I \in \mathbb{N}$ is the number of input channels, $k \in \mathbb{N}$ is the kernel size, $w^j \in \mathbb{R}^{I \times k}$ the kernels of the j -th output channel. This convolution is performed with zero padding, meaning that $x_t = 0$ for $t < 0$. To compute the output at time t the model uses the measurements $(x_{t-k+1}, x_{t-k+2}, \dots, x_t)$ which are all anterior to t , this is a way to model causality [18].

In [3], the CNN is based on two-dimensional convolution (**Conv2D**) defined as

$$y_t^{j,l} = \text{Conv2D}(x) = \sum_{i=0}^{d-1} \sum_{s=0}^{k-1} x_{t-\lfloor d/2 \rfloor + i}^{l-\lfloor k/2 \rfloor + s} w_s^{j,i},$$

where $w^j \in \mathbb{R}^{k \times d}$ are the kernels of j -th output channel of size $k \times d$. There is one more exponent ' l ' on y because the convolution is also performed on the dimension of features.

In this paper, we propose a one-dimensional causal CNN with skip connections [19]. The complete architecture is detailed in Figure 4 and is similar to the one used in [6]. A skip connection creates a shortcut from an early layer to a later one, linking the input of a convolutional block straight to its output [19]. Since various layers of a neural network capture distinct feature "levels," skip connections assist in preventing a drop in performance as more layers are added [19]. The CNN

is trained with the MSE Loss. To accelerate the training, we also apply the weight normalization [20] to re-parameterize the weights of each convolutional layer.

Physical properties in CNN

- **Causality:** Aiming to perform real-time estimation, the CNN has to be causal [18]. Thus, the model will only use present and past measures.
- **Equivariance to time-translation:** As demonstrated in [21], CNNs are equivariant to translation in the dimension of convolution. Time-translation equivariance is an important property of physical systems, since the physical law does not evolve through time.
- **Insensitivity to input-feature order:** The order of features in x is arbitrary and has no physical meaning.

In causal Conv1D, the j -th output channel at time t is

$$y_t^j = \sum_{1 \leq i \leq I} \sum_{s=0}^{k-1} x_{t-s}^i w_s^{j,i}$$

where I is the number of input channels. If the input channel of x is permuted (by π) to

$$\tilde{x}_t^i = x_t^{\pi(i)},$$

then one can find an equivalent CNN with kernel $\tilde{w}_s^{j,i} = w_s^{j,\pi(i)}$ to have the same output y_t^j . This property does not hold in general if one uses the Conv2D layer.

Advantage of CNN over PR:

- **Capturing Local Patterns [22]:** CNNs are designed to capture local patterns and features within the data. In the context of time series, local patterns can represent short-term temporal dependencies, which are crucial for understanding the dynamics of the time series. PR, on the other hand, cannot effectively capture these local patterns as it does not use past measures for predictions.
- **Automatic Feature Extraction [22]:** During the training step, CNNs automatically learn the relevant features from time series. PR uses directly the measures from sensors, which might not capture all the relevant information in the time series.
- **Handling Non-Linear Relationships [22]:** Time series data often contains complex non-linear relationships between the input and output variables. CNNs with their multiple layers and non-linear activation functions can better model these non-linear relationships compared to the PR, which is inherently limited to polynomial relationships. The activation functions of our model are LeakyReLU [23] :

$$\forall x \in \mathbb{R}, \text{LeakyReLU}(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0.1x & \text{elsewhere} \end{cases}$$

PR has the advantage of being simple to train and easy to understand, as it only involves 13 parameters. On the other hand, CNN are considerably more complex, requiring the tuning of 2,700 parameters and involving more sophisticated computational processes.

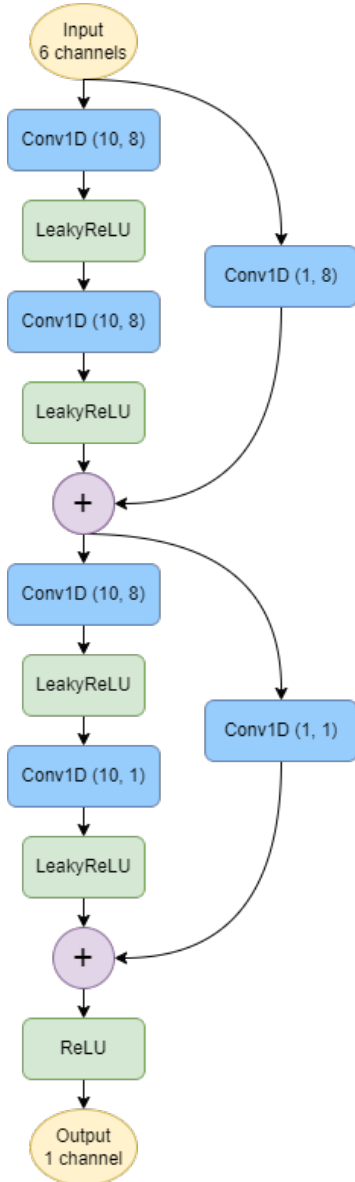


Fig. 4: Feedforward CNN
Conv1D (kernel size, number of output channels)

III. EVALUATION

EP Metrics: To evaluate the performance of the ML methods, a dataset of systems measures and their corresponding mass flow was split into training (64%), validation (18%), and testing (18%) sets. The training set was used to train the ML models, the model which had the smallest validation loss was selected, and the testing set was used to evaluate their performance. The performance of the ML models was evaluated using the MSE and engineering performance (EP) metrics that are standard in engineering fields and well-suited for analyzing dynamical behavior [5]. The EP metrics are defined as follows:

- $\Delta t_{80\%}, \Delta t_{10\%}$ [5]: 80% (resp. 10%) response time ($t_{80\%}, t_{10\%}$) is the time value at which the response signal has covered 80% (resp. 10%) of the ramp amplitude. $\Delta t_{80\%}$ (resp. $\Delta t_{10\%}$) is the absolute difference between

the true and predicted $t_{80\%}$ (resp. $t_{10\%}$)

- Δt_{peak} [5]: Peak delay is the absolute delay between the predicted signal and the target one.
- Δt_{conv} : for a signal containing a ramp followed by a static state, t_{conv} is the time when the signal has converged to the final value with 3% of tolerance. Δt_{conv} is the delay between the predicted and true t_{conv} .
- E_{abs} (resp. E_{rel}): Absolute (resp. relative) error on static states.

In order to compute a lot of characteristic times and static errors (EP metrics), a first step of segmentation was developed to identify the ramps ($\Delta t_{80\%}, \Delta t_{10\%}$, and Δt_{conv}), the overshoots and undershoots (Δt_{peak}) and the static states (E_{rel} and E_{abs}). Eventually, we have, for each of those metrics, several instances. To represent their distributions, we consider their 90% quantile.

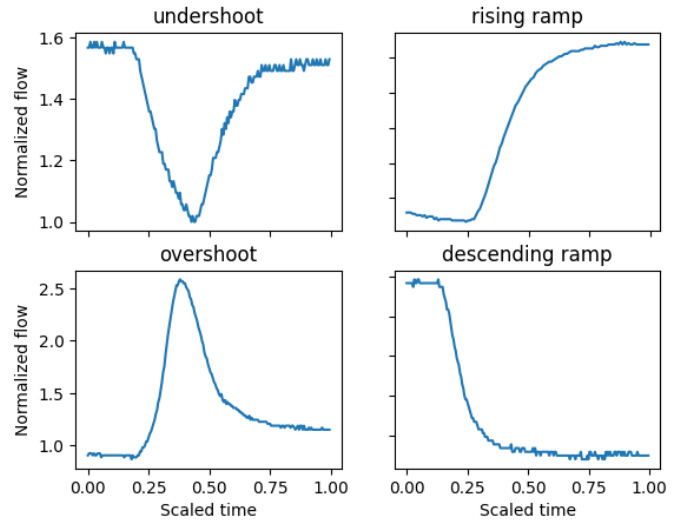


Fig. 5: Templates of patterns

The segmentation process: It consists of four steps. The patterns of interest are presented in Figure 5. The first step consists in selecting windows in the dataset that will constitute a set of candidates for being one of the patterns. For this purpose, a Butterworth [24] pass band filter of order 6 is applied to the flow with system-specific cutting frequencies. The frequency peaks are detected via the scipy function 'find_peaks' as shown in Figure 6. The candidates are symmetric windows around the detected peaks. Then, a Dynamic Time Warping (DTW) k-means clustering [25] with 6 clusters is performed on the normalized signals (Figure 7 in Appendix). DTW, while not a true distance measure, serves as an effective metric for clustering patterns. Specifically, two signals exhibiting identical patterns, albeit with a mere time delay, will manifest a small DTW value. This is evident in Cluster 2 of Figure 7, where even if overshoots happen at varying times, they remain closely aligned in terms of DTW. The identification of patterns is as follows:

- Cluster 1: Undershoots
- Cluster 2: Overshoots
- Cluster 3: Descending ramps

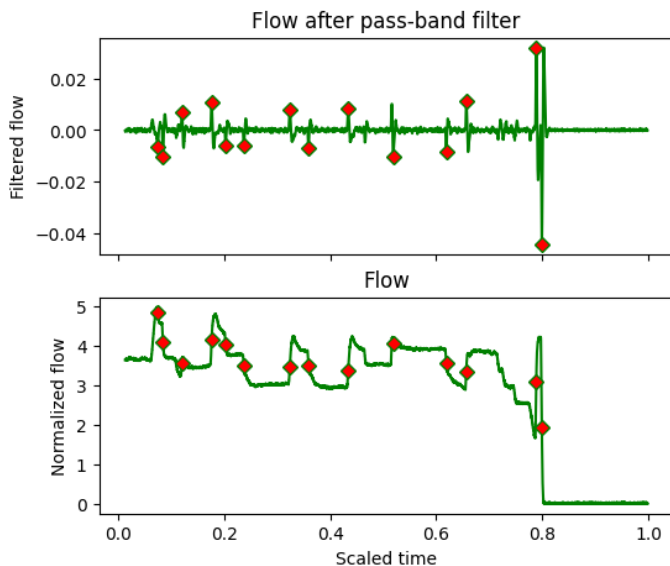


Fig. 6: First selection of candidates

- Cluster 4: Rising ramps

Cluster 5 and 6 correspond to other phenomenons like sinusoids. At this step, there are in clusters 1-4 some bad candidates. Aiming to remove those, we take the intersection with a DTW ball centered on the centers. A last manual selection is applied. Finally, we obtain satisfying numbers of each pattern (table II).

	train	test
Rising ramps	30	8
Descending ramps	35	5
Overshoots	79	12
Undershoots	22	8

TABLE I: Number of patterns detected in each set

IV. RESULTS

Our study at LIEBHERR presents its results in a normalized manner due to the sensitivity and protection of the VCS data. Even so, the obtained results can provide a clear comparison between the performance of PR and CNN models.

As tabulated in Table III, the Convolutional Neural Network (CNN) model demonstrates superiority over the PR model both in terms of Machine Learning (ML) metric (MSE) and Engineering Performance (EP) metrics. The CNN model particularly excels in predicting static errors and Δt_{conv} (convergence time).

As per the results, the CNN model outperforms the PR model by reducing the MSE from 0.18 to 0.06, the absolute error (E_{abs}) from 0.73 to 0.28, and the relative error (E_{rel}) from 28% to 14%. Furthermore, it provides a more accurate prediction of Δt_{conv} and $\Delta t_{80\%}$. PR has better results on $\Delta t_{10\%}$ and less good results on Δt_{peak} , however the differences are not significant. Thus, a better dynamic behavior is captured by the CNN model. Examples of predictions on patterns of interest can be found in Figure 8 in Appendix.

	PR		CNN	
	train	test	train	test
MSE	0.2	0.18	0.063	0.06
E_{abs}	0.53	0.73	0.19	0.28
E_{rel}	21%	28%	11%	14%
$\Delta t_{80\%}$	65	20	33	17
$\Delta t_{10\%}$	92	3	17	3.6
Δt_{conv}	97	44	74	14
Δt_{peak}	9	7.1	11	7

TABLE II: Comparison on MSE and 90% quantile

V. CONCLUSION

The study demonstrates that ML algorithms, specifically CNNs, can effectively function as virtual sensors in estimating physical quantities such as mass flow in aircraft compressor systems. Compared to PR, the causal one-dimensional CNN model, with the ability to capture dynamic behavior and non-linear relationships, outperforms in terms of MSE and EP metrics. The computation of multiple EP metrics on a large real dataset was performed via a semi-automatic segmentation method proposed in this article. The contribution is, on one hand, the design of a CNN-based model for estimating the flow in an industrial dataset from LIEBHERR Aerospace. On the other hand, the engineering evaluation is performed on several instances of ramps, overshoots, and undershoots.

Future work will focus on improving these models by using physical knowledge. For the virtual sensor of mass flow, the author of [2] proposed to use an incomplete equation (\star). There exists multiple other methods to integrate physical knowledge into ML Models [26].

REFERENCES

- [1] D. Martin, N. Kühn, and G. Satzger, "Virtual sensors," *Business & Information Systems Engineering*, vol. 63, 06 2021.
- [2] W. Kim, "Evaluation of virtual refrigerant mass flow sensors," *INTERNATIONAL REFRIGERATION AND AIR CONDITIONING CONFERENCE*, 2012.
- [3] H. Wan, "Development of dynamic modeling framework using convolution neuron network for variable refrigerant flow systems," *INTERNATIONAL REFRIGERATION AND AIR CONDITIONING CONFERENCE*, 2021.
- [4] L. Zhu, J. Liu, J. Liu, W. Zhou, D. Yu, J. Sun, F. Li, and J. Wan, "Frequent pattern extraction based on data and prior knowledge fusion in gas turbine combustion system," in *2017 Prognostics and System Health Management Conference (PHM-Harbin)*, 2017, pp. 1–8.
- [5] S. Verma, N. Henwood, M. Castella, A. K. Jebai, and J.-C. Pesquet, "Neural speed-torque estimator for induction motors in the presence of measurement noise," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 1, pp. 167–177, 2023.
- [6] J. Franceschi, A. Dieuleveut, and M. Jaggi, "Unsupervised scalable representation learning for multivariate time series," *CoRR*, vol. abs/1901.10738, 2019.
- [7] L. Alessandrini, M. Basso, M. Galanti, L. Giovanardi, G. Innocenti, and L. Pretini, "Maximum likelihood virtual sensor based on thermo-mechanical internal model of a gas turbine," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 3, pp. 1233–1245, 2021.
- [8] C. Arora, *Refrigeration and Air Conditioning*, ser. McGraw-Hill International editions: Mechanical technology series. Tata McGraw-Hill, 2000.
- [9] N. Jain and A. G. Alleyne, "Thermodynamics-based optimization and control of vapor-compression cycle operation: Optimization criteria," in *Proceedings of the 2011 American Control Conference*, 2011, pp. 1352–1357.

- [10] J. Zabczyk, *Mathematical Control Theory: An Introduction*, ser. Systems & Control: Foundations & Applications. Springer International Publishing, 2020.
- [11] R. Massiquet, "Compressor surge: Simulation, modeling and analysis," Ph.D. dissertation, 2022.
- [12] R. C. Baker, *Coriolis Flowmeters*, 2nd ed. Cambridge University Press, 2016, p. 560–602.
- [13] R. C. Baker, *Venturi Meter and Standard Nozzles*, 2nd ed. Cambridge University Press, 2016, p. 163–176.
- [14] S. L. Dixon, *Fluid mechanics and thermodynamics of turbomachinery*, 4th ed. Butterworth-Heinemann, 1998.
- [15] B. D. Rasmussen and A. Jakobsen, "Review of compressor models and performance characterizing variables," 2000.
- [16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–44, 05 2015.
- [17] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [18] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," 2016.
- [19] D. Wu, Y. Wang, S.-T. Xia, J. Bailey, and X. Ma, "Skip connections matter: On the transferability of adversarial examples generated with resnets," 2020.
- [20] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," 2016.
- [21] L.-H. Lim and B. J. Nelson, "What is an equivariant neural network?" 2022.
- [22] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015.
- [23] A. L. Maas, "Rectifier nonlinearities improve neural network acoustic models," 2013.
- [24] S. Butterworth, "On the Theory of Filter Amplifiers," *Experimental Wireless & the Wireless Engineer*, vol. 7, pp. 536–541, Oct. 1930.
- [25] F. Petitjean, A. Ketterlin, and P. Gancarski, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern Recognition*, vol. 44, pp. 678–, 03 2011.
- [26] C. Meng, S. Seo, D. Cao, S. Griesemer, and Y. Liu, "When physics meets machine learning: A survey of physics-informed machine learning," 2022.

APPENDIX

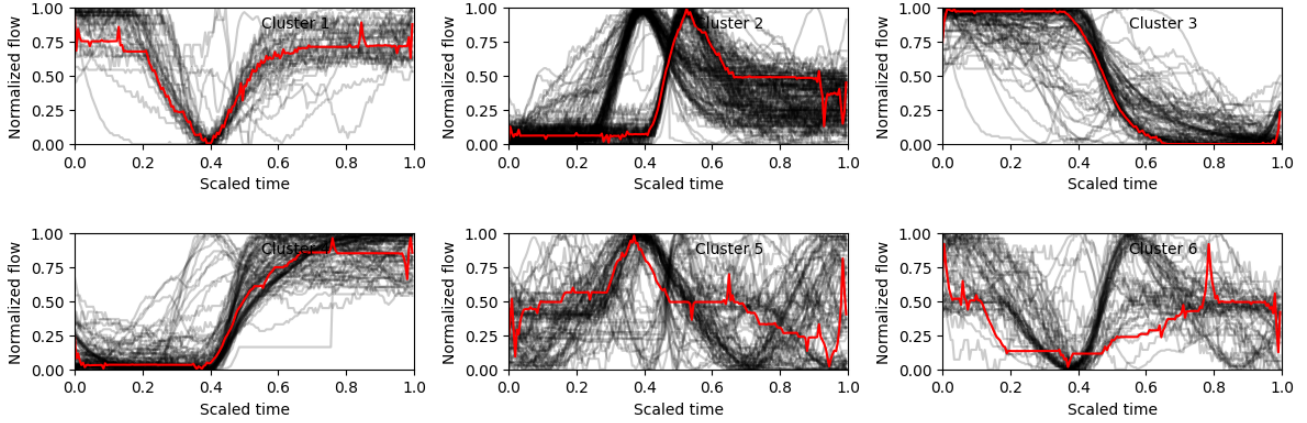
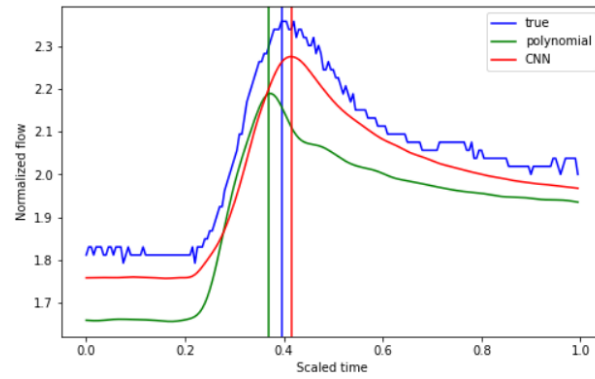
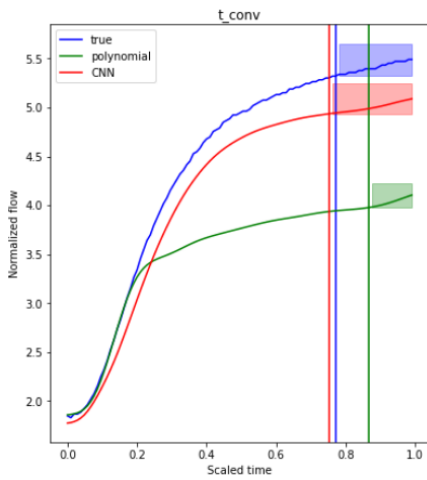


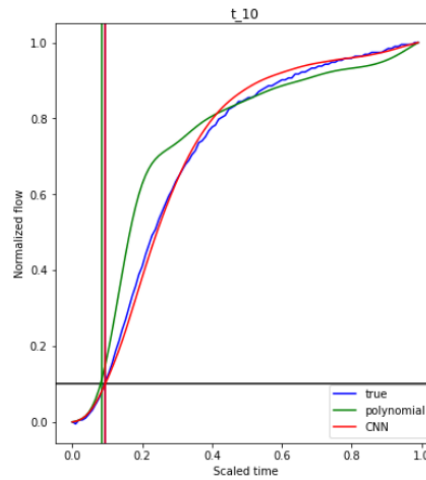
Fig. 7: Clusters with the DTW K-means algorithm. The centers are represented in red.



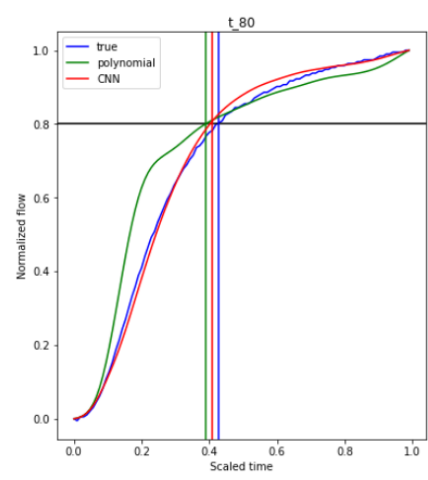
(a) Δt_{peaks}



(b) Δt_{conv}



(c) $\Delta t_{10\%}$



(d) $\Delta t_{80\%}$

Fig. 8: On (a) Δt_{peaks} is computed from an overshoot. In (b) Δt_{conv} is computed from a rising ramp with the two predictions. On (c) and (d) there is the same ramp as (b) but the three signals are scaled in order to have the same initial and final value. In this manner it is easier to compute $\Delta t_{10\%}$ and $\Delta t_{80\%}$.