

# Deep Learning vs. Classical Modeling of Processes for Fault Detection in Industrial Heating-Cooling Systems

Klaus René Garcia Rosas<sup>1,2</sup>, Martin Zimmer<sup>1</sup> and Bernhard Nebel<sup>2</sup>

<sup>1</sup>Fraunhofer Institute for Solar Energy Systems ISE

<sup>2</sup>Albert-Ludwigs-Universität Freiburg

e-mail: klaus.rene.garcia.rosas@ise.fraunhofer.de

## Abstract

Faults in heating-cooling systems can often be observed by changes in temperature. Such faults can be detected and identified by modeling thermodynamic behavior. In classical models, physical equations with fixed or trainable parameters are used to model this behavior. They are limited in non-linear complexity and the number of parameters to be estimated. They also usually require the involvement of expert knowledge. In this paper, a deep learning approach is presented for modeling thermodynamic behavior without explicitly modeling the physical properties. The modeled artificial neural network (ANN) can predict the temperature based on other influencing variables. A comparison with a mathematical-physical model (MM) shows that the ANN can reproduce temperature changes similarly good when sufficiently data is available. With increasing prediction windows, the ANN even outperformed the MM model for most states. Both models can detect certain heating faults by comparing the measured and predicted temperatures. Finally, we demonstrate the diagnostic capabilities of our methods by injecting a fault into the system.

## 1 Introduction

In the last decade, many industrialized nations like Germany have adopted strategic plans that view the digitization of the manufacturing industry as an essential economic competitive advantage in the future [1]. The automotive industry spends almost 200 billion US dollar a year in semiconductors to further automate manufacturing processes and products [2]. Although most industrial companies in Germany have already dealt with artificial intelligence (AI), only a fraction have implemented AI solutions for industrial processes [3]. The potential of AI is still growing, particularly through the development of special chips for training neural networks and the availability of user-friendly libraries. AI methods can help to ease complex modeling processes when enough heterogeneous data is available. They can help to facilitate fault detection, diagnostics and troubleshooting in plants which are essential to reduce maintenance time and faulty production. In this paper, the use of neural networks for fault detection in heating-cooling systems is compared with a classical mathematical model. In addition, the use of the models for comprehensive diagnostics is discussed.

## 2 State of the Art

Data-driven anomaly detection and fault detection relate in the way that data is used to model interdependent relationships between actuator and sensor values in a plant. But both detection methods differ in the way these relationships are represented and used. Anomaly detection mainly focus on general detection of rare observations in data, independent of component failures. Fault detection is more concerned to specifically detect component failures by modeling the physical relationships between actuators and sensors. Deep learning methods have been extensively used to perform anomaly detection tasks. For fault detection and diagnosis, deep learning methods were rarely applied when no fault labels exist. We therefore discuss both, anomaly and fault detection methods and their diagnostic capabilities. Anomaly detection methods for time-series data are mostly based on predicting data and compare the predicted values with the measured values. Besides of classical time prediction methods like ARIMA [4], various deep learning approaches exist. RNN-based methods, CNN-based methods and combinations of both have been used to predict univariate [5; 6] and multivariate time-series [7]. In general, anomaly detection works better when using multivariate time-series methods instead of univariate methods. However, the anomalies are harder to interpret for multivariate time-series methods. Zhang et al. use so called Autoencoders to combine anomaly detection for multivariate time-series and the identification of time-series groups that exhibit unusual behavior [8]. A further analysis to find the real cause or to determine error sizes is not possible with the methods described so far. A few applications exist where deep learning methods have been applied to build models for fault detection purposes without having fault labels. Helbing and Ritter list some approaches which use neural networks to predict temperatures in wind turbines [9]. Jiang and Maskell combine ANN and analytical based methods to detect faults and identify fault types for photovoltaic systems [10]. Most of these approaches do not consider time-dependencies, which is crucial to model time shifted effects. In this paper, we focus on deep learning methods to build fault detection models for time-dependent processes without existing fault labels.

## 3 Research Gap and Objectives

Component degradation and failure usually occur in plants only after a longer period of time. The behavior of time-series in the event of a fault can therefore not be modeled

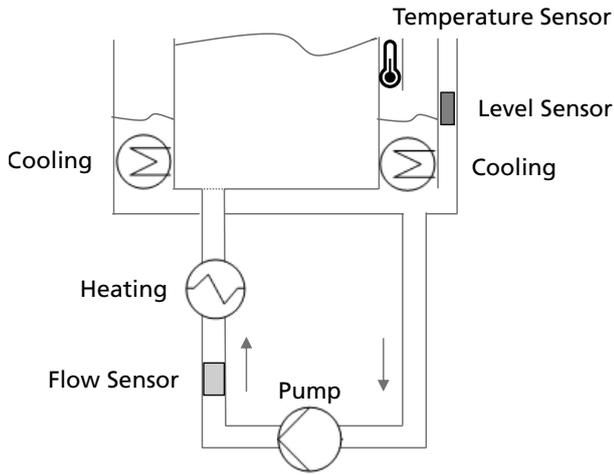


Figure 1: Main components of the station: the basin consisting of inner bath and overflow collar, the circulation system, the heater and the cooling system.

without expert knowledge or previously occurred faults. Although the problem appears to be one that could be solved by unsupervised learning techniques, it can also be seen as a supervised prediction problem in an unsupervised setting. This is done by using each time-series value as a prediction label for data recorded immediately before. Deep learning has become state of the art for anomaly detection tasks in unsupervised settings. To use it for fault detection and further diagnostic tasks, the model must be trained to learn the physical dependencies between dependent and independent variables. Several criteria can serve as benchmark for a good diagnostic model, such as high accuracy, robustness and interpretability. In the following, we explore the question of whether an ANN model is able to predict temperature values so precisely to replace a physical model for fault detection of heating and cooling failures. Furthermore, the usefulness for fault isolation and identification will be discussed to get a feel for the use of ANNs as replacement for physical models.

## 4 Wet-Chemical Processing System

Our test object is a batch processing system for the wet-chemical treatment of solar cells. The system consists of several stations for etching, cleaning, rinsing and drying wafers. We focus on etching stations where heating and cooling play a crucial role. Predefined substance amounts and concentrations are used for each process. Amounts and concentrations vary slightly with the number of process runs, but are balanced before each new run. An etching station consists of several components. The main ones are illustrated in Figure 1. The wet-chemical treatment of wafers take place in the inner bath of the basin. The overflow collar collects the overflow from the inner bath. A solution volume of 125l covers a complete filling of the inner bath and enough liquid to keep the circulation pipes full of liquid during circulation. A pump feeds the liquid from the overflow collar back to the inner bath. The inline heater is located before the liquid flows into the inner bath. The heater consists of three heating elements with a total power of 12kW. The cooling system consists of a water cooling pipe located in the overflow collar. The cooling pipe supplies water with

a constant temperature. The cooling water does not flow into the bath, but through a closed cooling circuit. Different sensors are installed: a temperature and a level sensor at the transition between inner bath and overflow collar, a level sensor in the overflow collar and a flow sensor in the circulation. We further get switching signals (on/off) for the heater, the cooling system and the pump.

## 5 Methodology

Two models are compared: a traditional model described by physical equations and a deep learning model comprising a convolutional neural network. Both are designed to predict the solution temperature by several influencing factors.

### 5.1 Mathematical-Physical Model (MM)

The mathematical-physical model (*MM model*) serves as reference model. It is based on several laws of thermodynamics. We mainly model three influencing components and their effects on the solution temperature: heating, active cooling and passive cooling. Passive cooling describes the cooling process caused by the ambience. Passive cooling is always present, while heating and active cooling are only active when the heating and cooling system are turned on, respectively.

#### Passive Cooling

Passive cooling is modeled by Newton's law of cooling. The law states that the rate of heat loss of a body is proportional to the difference in temperatures between the body  $T$  and its ambience  $T_{env}$ :

$$\frac{dQ}{dt} = -k_{passivecooling}(T - T_{env}) \quad (1)$$

$k_{passivecooling}$  is the decay constant and  $T_{env}$  the ambient temperature.

#### Active Cooling

The active cooling is also modeled by Newton's law of cooling. Similar to passive cooling, the rate of heat loss can be described by

$$\frac{dQ}{dt} = -k_{activecooling}(T - T_{cool}) \quad (2)$$

$k_{activecooling}$  is the decay constant and  $T_{cool}$  the temperature of the cooling water.

#### Heating

The heating is modeled as heat source with lossless heat transfer. We assume the rate of heat gain as constant, so that

$$\frac{dQ}{dt} = \Delta T_{heating} \quad (3)$$

#### Temperature Change

The temperature change of the solution can be calculated using the heat energy formula  $Q = C \cdot \Delta T$  where  $Q$  is the heat energy and  $C$  the heat capacity of the solution. Due to the hardly varying concentration of the chemicals,  $C$  is nearly proportional to the volume  $V$  of the solution. To model the heating and cooling constants dependent on  $V$ , the temperature change of the solution can be described by

$$\Delta T = \frac{Q}{V} \quad (4)$$

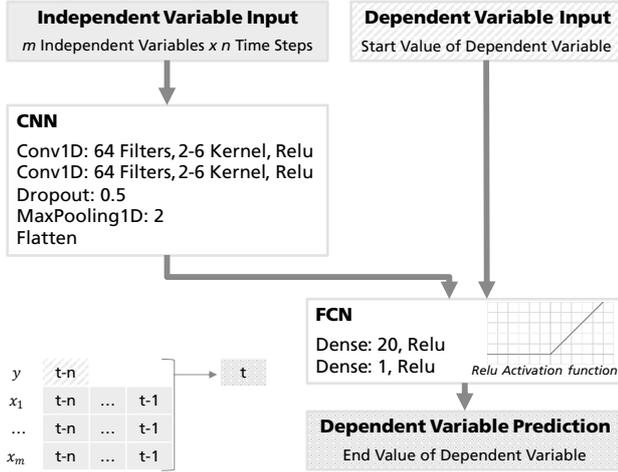


Figure 2: ANN structure. Values of independent variables from time  $t-n$  to  $t-1$  are fed into the CNN. Together with the start value of the dependent variable at time  $t-n$ , the results of the CNN serve as input for the FCN. The output is the final prediction of the dependent variable at time  $t$ .

### Final Formula

Putting it all together, we get the formula

$$\begin{aligned} f_h(t) &= H(t)\Delta T_{heating} \\ f_{ac}(t) &= K(t)(-k_{activecooling}(T - T_{cool})) \\ f_{pc}(t) &= -k_{passivecooling}(T - T_{env}) \end{aligned} \quad (5)$$

$$T(t) = T(t-1) + \frac{f_h(t-1) + f_{ac}(t-1) + f_{pc}(t-1)}{V(t-1)} \quad (6)$$

with variables

$H(t)$	0: heating is off at time $t$ 1: heating is on at time $t$
$K(t)$	0: active cooling is off at time $t$ 1: active cooling is on at time $t$
$T(t)$	temperature of solution at time $t$
$V(t)$	volume of solution at time $t$
$T_{cool}$	temperature of cooling water
$T_{env}$	ambient temperature
$\Delta T_{heating}$	heating temperature change constant
$k_{activecooling}$	decay constant of active water cooling
$k_{passivecooling}$	decay constant of passive cooling

We assume that the cooling water is constantly at  $18^\circ\text{C}$  and the ambient temperature constantly at  $23^\circ\text{C}$ .

### 5.2 Artificial Neural Network Model (ANN)

Our artificial neural network model (*ANN model*) is a special combination of a convolutional neural network (CNN) and a fully connected neural network (FCN). The structure is visualized in Figure 2. A convolutional neural network is used to handle time dependencies between variables. In contrast to convolutional neural networks commonly used for anomaly detection tasks where all the input is fed into the CNN, the data is split into one *dependent* variable and the corresponding *independent* variables. The goal is to model the dependent variable to be predicted not by its previous values, but the previous values of the independent variables. With this approach, we hope to achieve a better modeling of the real influencing factors. In our case, temperature is the

dependent variable. Heating, cooling and volume are the independent variables. We would like to predict the temperature at time  $t$  using the temperature at time  $t-n$  and all independent variables from time  $t-n$  until  $t-1$ . If we would also use the temperature from time  $t-n+1$  to  $t-1$  to predict the temperature at time  $t$ , we could just use the temperature at time  $t-1$  to get a very close approximation. But our goal is that the model learns to predict temperature changes based on the influencing variables. For this reason, we don't include the intermediate temperature values. The convolutional neural network itself consists of several layers. The input layer takes all independent variables for several time steps as a two-dimensional matrix. Two consecutive one-dimensional convolutional layers follow. Each layer is using a kernel size of  $\max(2, \min(\frac{n}{3}, 6))$  and 64 filters. Relu is used as activation function [11]. Two more layers follow: a dropout layer and a MaxPooling1D layer. Both layers are used to avoid overfitting. The flatten layer flattens the resulting two-dimensional matrix into a single vector. This vector is used together with the dependent variable at time  $t-n$  as the input for the FCN. The FCN consists of two fully connected layers. In the first one, the input vector is fully connected to 20 nodes. After the second one, we get the predicted dependent variable at time  $t$ . For both layers, the Relu activation function is used.

### 5.3 Model Parameters

The *mean squared error (MSE)* serves as the loss function for both models to optimize the parameters. The Levenberg-Marquardt optimizer [12] is used for the MM model. For the ANN model, the Adam optimizer [13] is used, and the batch size is set to 100. To avoid overfitting in the ANN, 20% of the training set is used as validation set. Training is stopped as soon as there is no improvement in performance on the validation set within 5 epochs.

## 6 Data Preprocessing

The wet-chemical plant provides data every second. Data from a rather shorter processing day are shown in Figure 3. The data includes the temperature, whether the heating is on or off, whether the active cooling is on or off, the level of the solution and whether the circulation is on or off. The volume of the solution is calculated by the level and circulation value. Training and testing data cover a total of three months. Active processing took place throughout the day, five days a week. The data is aggregated to 10 s. Mean is used as aggregation function for temperature and level. Relative frequency of activity is used for heating, active cooling and circulation. To generate the model input data, a rolling time window of size  $w$  is used, which consists of  $w$  consecutive time steps of 10 s.

## 7 Results

The study is carried out in three steps. First, the performance of the models is compared for different assumptions over the time delays between actuators and sensors. Secondly, the performance is compared for different prediction windows. In the last step, relative prediction errors of the temperature changes for different states are investigated to evaluate the potential use for further diagnostic purposes. Unless otherwise specified, a *3-fold cross-validation (3-CV)* is used to evaluate performance. The data set is split only by day to prevent tearing apart too much consecutive data.

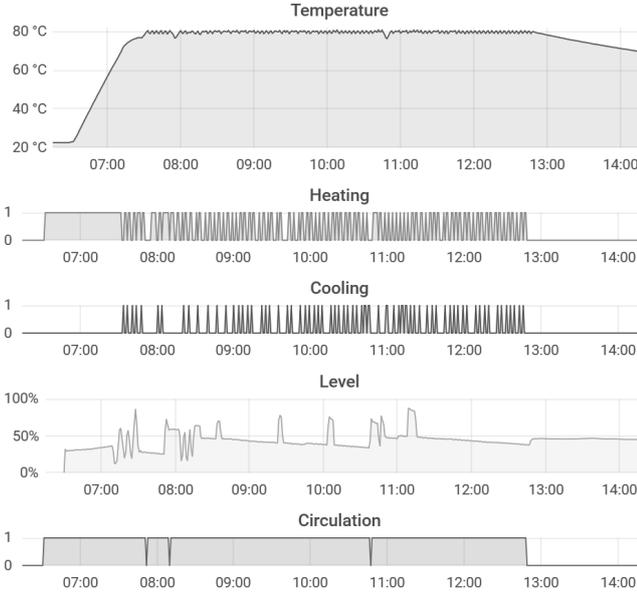


Figure 3: Before processes take place, the circulation is turned on and the solution is heated. After the desired temperature has been reached, the bath is set for processes. The solution is kept at temperature by alternating heating and cooling phases. The level changes especially during processing. After all processes of the day have taken place, the heating, the cooling and the circulation are switched off.

State	Count	Rel. Count in %	$\Delta T$ Mean in $^{\circ}\text{C}$	$\Delta T$ Std in $^{\circ}\text{C}$
Off	651k	85.7	-0.02	0.07
Cool	4k	0.5	-0.50	0.30
Heat	34k	4.4	0.78	0.60
Mixed	72k	9.4	-0.12	1.05
Total	760k	100.0	0.00	0.39

Table 1: States for  $w=6$  without shifts. Counts refer to the number of windows.  $\Delta T$  are temperature differences between end and start temperatures.

In the MM model, the temperature prediction is done sequentially, time step by time step. This means that predicted temperatures of intermediate time steps are used as input for the next temperature prediction until the final temperature is predicted.

### 7.1 Time Delays

The performance for varying time delays for heating and cooling is investigated for both models. A prediction window of 1 min is selected for this analysis. The numbers of different states when no time delays are considered can be looked up in Table 1. With 85.7%, most of the time neither the heating nor the cooling is on (*Off*). Windows in which only the cooling is on continuously (*Cool*) are rare, with a relative share of 0.5%. In 4.4% of cases, only the heating is on continuously (*Heat*). In 9.4% of cases, either heating, cooling or both are on at any time within the time window (*Mixed*). The latter state further excludes cases that already fall into state *Cool* or *Heat*. Time delays are examined in steps of 10 s ranging from 0 s to 70 s seconds for heating and cooling. This is done to determine the delay effect between switching on the heater or cooling and measuring the tem-

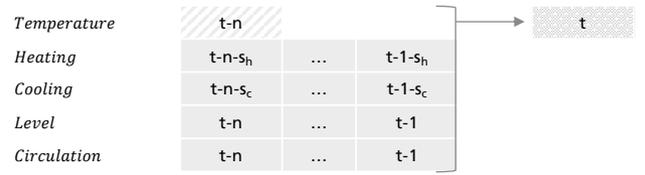


Figure 4: Vectors are shifted to the right by  $s_h$  steps for heating and  $s_c$  steps for cooling, while the other values stay in place.

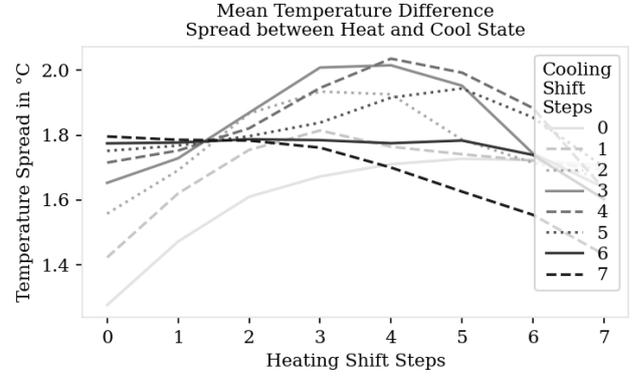


Figure 5: Spread of mean temperature differences between *Heat* and *Cool* state.

perature. Time delays are applied to the input data by shifting the heating and cooling vectors by the time delay, like visualized in Figure 4. When shifts are applied, the states are also reclassified. For example, if a heating time delay of 10 s is assumed and the heating is switched on after a longer phase without heating and cooling, the subsequent 10 s are still classified as *Off* states. Only after the 10 s a heating effect is assumed and the *Off* state changes to a *Mixed* or *Heat* state depending on  $w$ . In the following, the influence on the temperature difference spread between *Heat* and *Cool* states is examined. Afterwards, the general performance is evaluated for both models using the *root-mean-squared error (RMSE)* and the *mean-absolute error (MAE)* as metrics.

### Temperature Difference

We expect the temperature to raise if the heating is on and fall if the cooling is on. The spread between the mean temperature difference of state *Heat* and state *Cool* serves as indicator of how well the delay effect of heating and cooling is mapped. This is because we expect higher average temperature changes for *Heat* states when the actual heating periods are met. The equivalent holds for *Cool* states. For  $w=6$  without shifts, the spread amounts to  $0.78^{\circ}\text{C} - (-0.50^{\circ}\text{C}) = 1.28^{\circ}\text{C}$ . The spreads for different time delays for cooling and heating are visualized in Figure 5. With  $2.04^{\circ}\text{C}$  the largest spread was achieved for a heating shift of 4 steps and a cooling shift of 4 steps. This equals 40 s time delay for both heating and cooling.

### RMSE

For the RMSE, the squared errors are summed up over the 3 CV runs before applying the root mean. Results are shown in Figure 6. First, the results for the MM model are considered. Looking at each cooling shift curve separately, the best result is between 2 and 5 heating shift steps. Looking at each heating shift curve separately, the best cooling

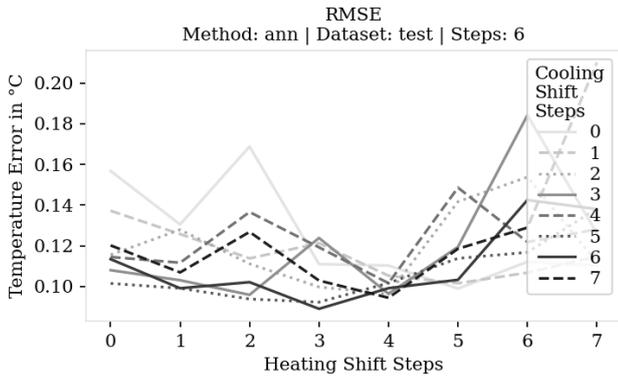
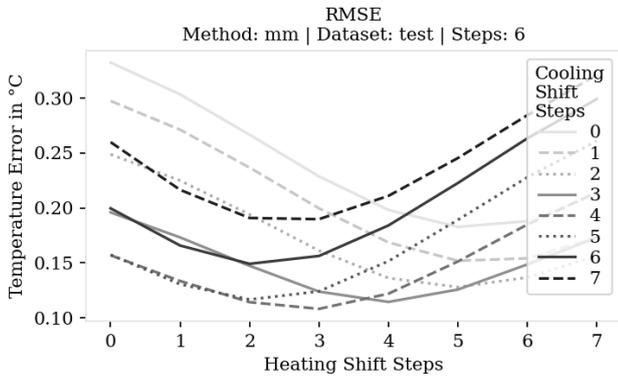


Figure 6: RMSE for different heating and cooling shifts.

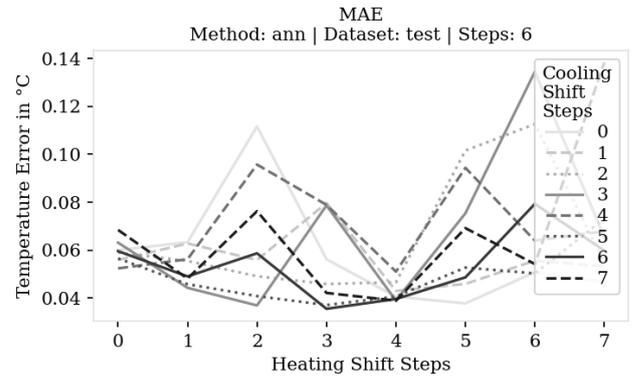
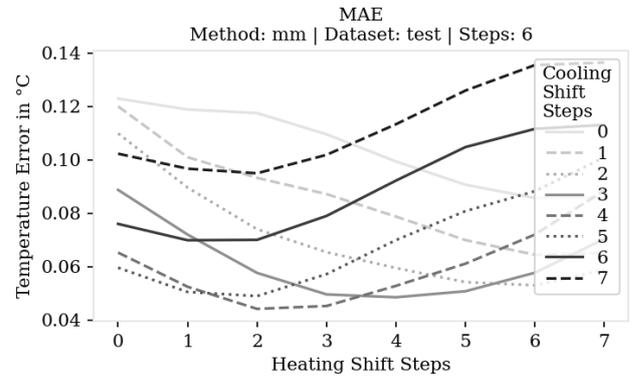


Figure 7: MAE for different heating and cooling shifts.

shift result is also between 2 and 5 steps. The best result with an RMSE of  $0.108\text{ }^{\circ}\text{C}$  is achieved by setting the heating shift to 3 steps and cooling shift to 4 steps. This equals 30 s time delay for heating and 40 s time delay for cooling. For the ANN model, looking at each cooling shift curve separately, the best result is also between 2 and 5 heating shifts. Looking at each heating shift curve separately, there is no clear best result for the cooling shift. The best result with an RMSE of  $0.093\text{ }^{\circ}\text{C}$  is achieved by setting the heating shift to 3 steps and cooling shift to 6 steps. This equals 30 s time delay for heating and 60 s time delay for cooling. The ANN model performed better than the MM model in most cases, but the results are less clear for the ANN model. Deriving the true time delay for heating and cooling does not appear to be straightforward using the ANN model.

### MAE

For the MAE, the absolute errors are summed up over the 3 CV runs before applying the mean. Results are shown in Figure 7. The best result for the MM model is achieved by setting the heating shift to 2 or 3 steps and the cooling shift to 4 steps. This corresponds to 20 s or 30 s time delay for heating and 40 s time delay for cooling. Both configurations led to a MAE of  $0.048\text{ }^{\circ}\text{C}$ . The best result for the ANN model is achieved by setting the heating shift to 3 steps and the cooling shift to 6 steps. This configuration led to a MAE of  $0.038\text{ }^{\circ}\text{C}$ . This corresponds to 30 s time delay for heating and 60 s time delay for cooling.

## 7.2 Time Window Size

In this subsection, the performance for varying prediction windows is investigated in detail. Window sizes of 1 min, 2 min and 5 min are compared. In the previous subsection, the best result for the 1 min MM model was achieved by

State   $w$	Rel. Count in %			$\Delta T$ Mean in $^{\circ}\text{C}$		
	6	12	30	6	12	30
Off	85.7	85.2	84.8	-0.03	-0.05	-0.11
Cool	0.44	0.08	0.02	-0.76	-1.27	-5.00
Heat	4.2	1.6	1.3	1.19	2.47	6.25
Mixed	9.6	13.1	13.8	-0.27	0.01	0.08

Table 2: Counts and temperature changes by state and window size applying 3 heating and 4 cooling shift steps.

shifting the heating by 3 steps and cooling by 4 steps. The same process is repeated for a 2 min and 5 min prediction window, leading to the identical best configuration. A 30 s time delay for heating and a 40 s time delay for cooling is therefore applied to the input data for all MM models. For the 1 min ANN model, a 30 s time delay for heating and a 60 s time delay for cooling performed best. For the 2 min and 5 min prediction window, a 30 s time delay for heating and 40 s time delay for cooling performed best. These configurations are applied to the input data for the corresponding ANN models.

The numbers of different states for 1 min, 2 min and 5 min windows of consecutive data are shown in Table 2. About 85 % of the time, neither the heating nor the cooling is on for all window sizes. Dependent on the window size, the proportions for the other states vary a lot. The proportion of mixed heating and cooling logically increases with the window size, while the exclusively cooling and exclusively heating states decrease. Having 0.44 % of time steps falling into cooling states for 1 min windows, only 0.02 % of time steps remain when using 5 min prediction windows. Likewise, heating states drop from 4.2 % to 1.3 % comparing the 1 min and 5 min window, respectively. In contrast, mixed heating and cooling states increase from 9.6 % to 13.8 %.

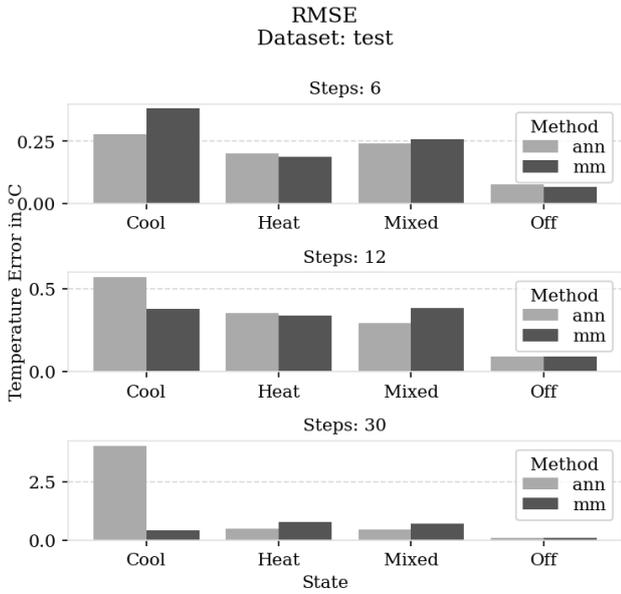


Figure 8: RMSE for different states and models considering prediction window sizes of 1 min, 2 min and 5 min.

Two metrics are used to evaluate the performance for different prediction windows and states. First, the RMSE is used to analyze the absolute temperature error. Afterwards, the median of the absolute relative temperature change error is used to evaluate the performance in terms of relative temperature changes.

### RMSE

For the RMSE, the squared errors are summed up over the 3 CV runs before applying the root mean. Figure 8 shows the results for different prediction windows. Looking at the Cool states, the ANN model performs better than the MM model for the 1 min prediction window, while the MM model performs better for the 2 min and 5 min windows. Looking at the Heat states, the MM model performs slightly better for the 1 min and 2 min prediction windows, while the ANN model performs better for the 5 min window. Looking at the Mixed states, the ANN model outperforms the MM for all window sizes. Looking at the Off states, both models are almost on par for all window sizes. With increasing window size, the ANN model gains performance compared to the MM model for Heat, Mixed and Off states.

### Relative Error in Temperature Change

Relative temperature change errors can be used to relate the absolute prediction errors to the real temperature changes. The absolute relative temperature change error is defined by

$$\left| \frac{\Delta T_{prediction} - \Delta T}{\Delta T} \right| \quad (7)$$

where  $\Delta T_{prediction}$  is the predicted temperature change of a model and  $\Delta T$  the measured temperature change. All absolute relative errors of the 3 CV runs are considered to calculate the median for each combination of method, state and window size. Values with a zero denominator are ignored. Results are visualized in Figure 9. For Cool states, the ANN model has high median errors between 40 % and 73 %. The MM model performs better, with median errors between 9 % and 28 %. For Heat states, median errors are between

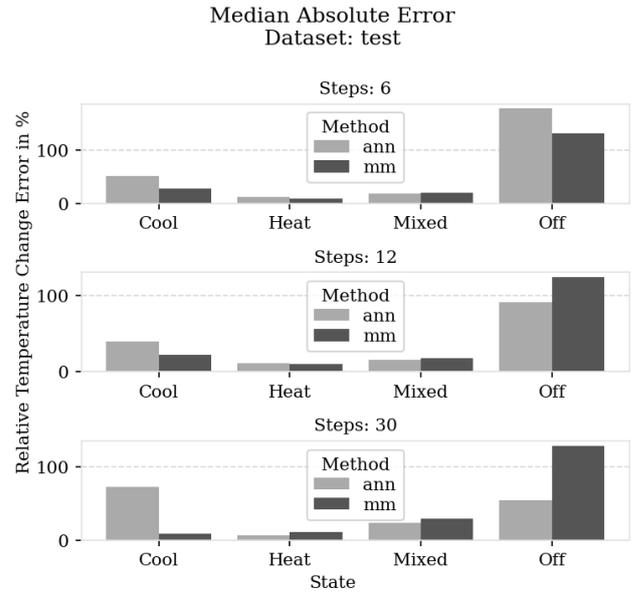


Figure 9: Median of absolute relative temperature change errors for different states, models and window sizes.

9 % and 12 % for the 1 min and 2 min windows. Only for the 5 min window, the ANN model outperforms the MM model with a median error of 7 % compared to 11 %, respectively. For Mixed states, the ANN model achieves better results than the MM model. Median errors are between 16 % and 30 % for all methods and window sizes. Median errors for the Off states are significantly higher than for all other states. While the errors reach values above 100 % for all window sizes of the MM model and the 1 min ANN model, errors decrease for the ANN models with increased window size. With 54 %, the best result is achieved by the ANN model for the 5 min window.

### 7.3 Detection of Heating Losses

In this subsection, the role of relative prediction errors for diagnostic tasks is discussed. Further, we demonstrate the diagnostic capabilities by provoking a failure.

#### Relative Power

Relative prediction errors in terms of temperature changes are useful to evaluate heating and cooling losses. A heating power loss can be detected, for example, by a smaller temperature increase than expected. This can be well quantified as the relative power calculated by

$$P_{rel} = \frac{\Delta T}{\Delta T_{prediction}} \quad (8)$$

If the measured change in temperature equals the predicted change,  $P_{rel}$  would be 1 which corresponds to 100 % power. When the heater is no longer providing full power,  $P_{rel}$  would drop below 1 as the expected change in temperature would be higher than the measured change. In Figure 10, histograms for calculated  $P_{rel}$  are shown for a 2 min prediction window. Values lower than 50 % were set to 50 %. Values higher than 200 % were set to 200 %. The relative deviation from the 100 % mark serves as the error metric. Looking at the Cool states, the ANN model underestimates and the MM model overestimates the real cooling effect in most cases. Accurate detection of cooling defects is unlikely

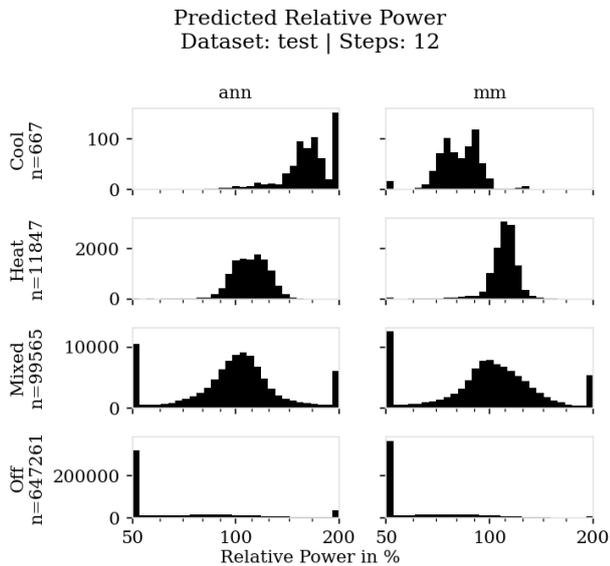


Figure 10: Number of predicted  $P_{rel}$  for a 2 min prediction window grouped by state and method.

no matter which model is used. For heating states, the distribution is denser. Both models slightly underestimate the majority. Deviations from the 100 % mark might be low enough for both models to detect heating defects with a certain defect severity. For Mixed states, deviations from the 100 % mark are generally high. Nevertheless, errors might be detected depending on the frequency of heating and cooling phases. If the heating is on most of the time, detecting a heating loss is more likely than if it is off most of the time. For Off states, high deviations from the 100 % mark occurred frequently. Neither of both models can be used in this form to detect changes in passive cooling behavior.

### Test Case

The loss of heating power is demonstrated experimentally: One of the three heating elements of the heater is deactivated during a phase of alternating heating and cooling. Both models with a 2 min prediction window are tested. An alarm is triggered when the relative power for a heat state drops constantly below 80 % for 5 min. The results for the test case are illustrated in Figure 11. The regular initial heating procedure starts at 7:23. The desired temperature is reached at 7:59. Alternating cooling and heating states follow to keep the temperature around the desired temperature. A single etching process starts at 8:59 and ends at 9:06. At 9:16, one of the three heating elements is deactivated. The deactivated element is reactivated at 9:34. During the initial heating procedure, both models can predict the temperature change quite accurately. The relative power fluctuates slightly above 100 %. Additionally, during alternating cooling and heating states, the temperature change prediction seems to be quite accurate. During the etching process, the prediction gets worse, resulting in two relative power peaks for which one is far lower and the other far higher than 100 %. They only last a short time. The alarm is therefore not triggered. After manipulating the heater, the relative power drops to values which are constantly between 40 % and 60 % for both models. The alarm is triggered after 5 min. After reactivating the deactivated heating element, the calculated relative power increases again. The provoked

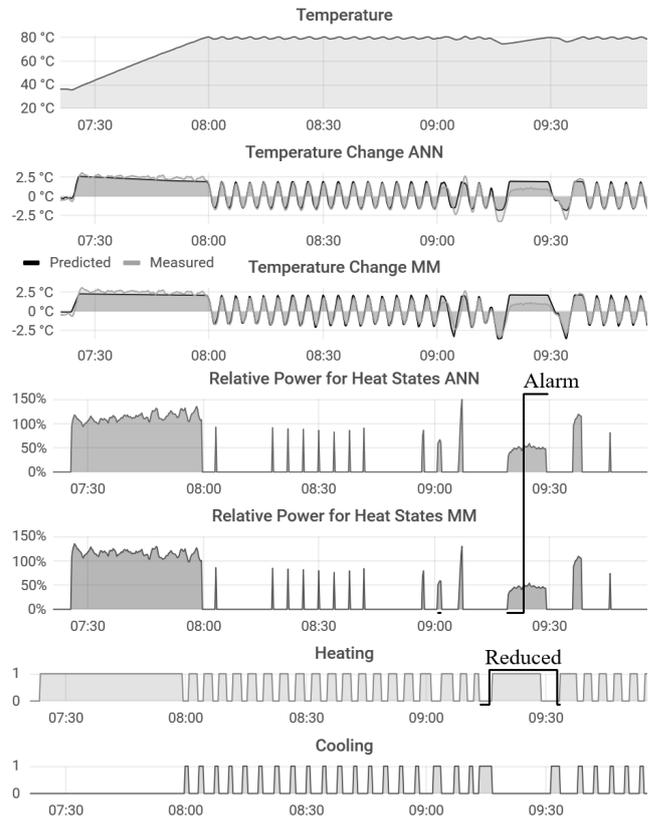


Figure 11: One heating element is deactivated at 9:16. Relative power drops sharply below 100 %, as the expected temperature increase is significantly higher than the real one.

heating failure was severe enough for a successful fault detection.

## 8 Discussion

Regarding the performance of both models, no clear winner could be determined. Both models performed best for Heat states, considering the relative temperature change error. For Cool states, relative errors of the ANN model are high. The most likely reason for the high errors is the small number of available data sets for cooling states. Increasing the number of cooling states may help to improve the performance. The ANN model generally performed better than the MM model for Mixed states. Looking at Off states, relative errors are high for both models.

Regarding fault diagnosis, both models can detect heating faults with sufficiently large severity. The magnitude of the fault can be quantified by the relative power loss. For the detection of active and passive cooling faults, both models are not suitable in this form. Besides the small number of training examples for active cooling states, likely reasons are the assumption over the ambient temperature and the high noise compared to the small temperature change for small prediction windows. We assumed the ambient temperature to be static but in particular when several processes take place in the plant, a significantly higher ambient temperature is likely. Underestimated cooling powers during active and passive cooling processes are the result. Measuring and using the real ambient temperature may improve the performance of both models. A larger prediction window may further be helpful to detect faults in passive cooling.

In addition to the influencing variables considered, there are still others that have an effect on the temperature but were not considered in the models. These variables are in particular the temperature of the incoming carriers and wafers, the reaction heat, the temperature of the added chemicals and different concentrations and amounts of chemicals in the process bath. The incoming carriers and wafers have lower temperatures than the solution at the time of processing, but the chemical reaction causes additional heating. Neither is easily quantifiable, but both have an impact on the temperature. The dosing of chemicals is done before initially heating the bath. Since temperatures are usually low at this time, the effect on the temperature is only minor and for a short time. Using minimum times for threshold exceedances before an alarm is triggered may prevent the detection of false positives caused by short time influences. Different concentrations and amounts of chemicals have an impact on the temperature as well. During processing, however, care is taken to ensure that the composition of the chemicals may change only slightly. The effect is therefore negligible. As still various minor influencing variables exist, they can be hardly modeled in a mathematical-physical model. The ANN model on the other hand does not need to specify equations for the thermodynamic behavior. Further, influencing variables can be easily included. Even an explicit time shift may not be necessary. Just extending the non-shifted matrix of independent variables by the values of previous time steps may be enough to consider the time delays of heating and cooling. The ANN by itself can consider time dependencies, which is the main reason it consists partially of a CNN. A comparison between ANNs with the shifted and the non-shifted extension version applied to the input data is still pending.

## 9 Conclusion

We showed that an ANN model can calculate temperature changes of a wet-chemical heating-cooling system by other influencing variables similarly good as a mathematical-physical model. Both models can detect certain types of faults. An induced heating failure was successfully detected by both models. We investigated weaknesses in both models and identified various approaches to improve the temperature prediction, fault detection and fault identification. Both models could be made useful for detecting cooling faults by using a sufficiently large set of cooling states and using the real ambient temperature instead of a static one. We showed that modeling the thermodynamic behavior in a classic way is limited, whereas models based on neural networks can include hard to model influences in a simpler way. Once the diagnostics have improved to the point where they are sufficiently good for common cooling and heating faults, other potential fault cases can be modeled until we have a diagnostic model for the main faults in the system.

## Acknowledgments

This research is funded by the Federal Ministry for Economic Affairs and Energy of Germany (BMWi) within the project CHEOPS (grant number 0324056B).

## References

- [1] BITKOM, VDMA, and ZVEI. Umsetzungsstrategie Industrie 4.0.
- [2] PwC. Opportunities for the global semiconductor market.
- [3] Wilhelm Bauer, Claudia Dukino, Michaela Friedrich, Walter Ganz, Moritz Hämmerle, Falko Kötter, and Stuttgart Fraunhofer IAO. *Künstliche Intelligenz in der Unternehmenspraxis: Studie zu Auswirkungen auf Dienstleistung und Produktion*. Fraunhofer Verlag, Stuttgart, 2019.
- [4] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time series analysis: Forecasting and control*. Wiley Series in Probability and Statistics. John Wiley & Sons, Hoboken, New Jersey, 5th edition, 2015.
- [5] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using LSTMs and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 387–395, New York, NY, USA, 2018.
- [6] Ya Su, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, and Dan Pei. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2828–2837, New York, NY, USA, 2019.
- [7] Mohsin Munir, Shoab Ahmed Siddiqui, Muhammad Ali Chattha, Andreas Dengel, and Sheraz Ahmed. FuseAD: Unsupervised anomaly detection in streaming sensors data by fusing statistical and deep learning models. *Sensors (Basel, Switzerland)*, 19(11), 2019.
- [8] Chuxu Zhang, Dongjin Song, Yuncong Chen, Xinyang Feng, Cristian Lumezanu, Wei Cheng, Jingchao Ni, Bo Zong, Haifeng Chen, Nitesh V. Chawla. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. *AAAI Proceedings*, 2019.
- [9] Georg Helbing and Matthias Ritter. Deep learning for fault detection in wind turbines. *Renewable and Sustainable Energy Reviews*, 98:189–198, 2018.
- [10] Lian Lian Jiang and Douglas L. Maskell. Automatic fault detection and diagnosis for photovoltaic systems using combined artificial neural network and analytical based methods. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. 2015.
- [11] R. H. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405(6789):947–951, 2000.
- [12] Jorge J Moré. The Levenberg-Marquardt algorithm: implementation and theory. In *Numerical analysis*, pages 105–116. Springer, 1978.
- [13] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015.