*Proceedings of the 2024 Winter Simulation Conference*
*H. Lam, E. Azar, D. Batur, S. Gao, W. Xie, S. R. Hunter, and M. D. Rossetti, eds.*

# COMBINING SIMULATION AND RECURRENT NEURAL NETWORKS FOR MODEL-BASED CONDITION MONITORING OF MACHINES

Alexander Wuttke[1], Markus Rabe[1], Joachim Hunker[2], and Jan-Philipp Diepenbrock[3]

[1]Dept. IT in Production and Logistics, TU Dortmund University, Dortmund, GERMANY
[2]Fraunhofer Institute for Software and Systems Engineering, Dortmund, GERMANY
[3]IVA Schmetz GmbH, Menden, GERMANY

## ABSTRACT

Maintenance is pivotal in industry, with condition-based maintenance emerging as a key strategy. This involves monitoring the machine condition through sensor data analysis. Model-based approaches compare observed data with expected values from models, which requires high-quality models. An established method is to use simulation models, which in many cases produce good results but may lack precision due to uncertainties. Alternatively, models created by machine learning can detect patterns directly from data. This paper proposes combining simulation models with machine learning models, leveraging the simulation's a-priori knowledge and machine learning's data patterns to enhance models for condition monitoring. Recurrent neural networks are suggested as the machine learning method. The paper outlines a systematic approach and demonstrates its application in an industrial use case, which investigates vacuum processes in industrial furnaces.

## 1 INTRODUCTION

Implementing an effective maintenance strategy for machines is a high priority in industry. This is due to its high share of running costs as well as the costs of unplanned downtimes, which can significantly disrupt the entire production operation (Isermann 2011). A promising implementation strategy is condition-based maintenance (CBM) (Quatrini et al. 2020). This strategy involves setting up a condition monitoring (CM) system based on data collected by sensors and deriving an assessment of the condition from these data. Ideally, CBM enables predictions on when machines will fail in the future and, therefore, allows for timely planning according to it. CM can be realized by model-based approaches. Such approaches leverage models to predict behavior and detect deviations from this occurring in the sensor data.

Models used for this purpose can be simulation models based on physical laws and mathematical equations. As a-priori knowledge is used to create these, they are also referred to as knowledge models. While these models can replicate machines effectively, they inevitably encounter aspects of reality that are challenging to model due to complexity or lack of understanding. Besides knowledge models, there are also data models, which do not (or only to a small extent) incorporate a-priori knowledge but knowledge that can be found in large amounts of data. Data models can be created by utilizing methods of machine learning (ML). Particularly, a method based on recurrent neural networks (RNNs) is expected to be especially suited to time series data (Goodfellow et al. 2016). A limitation of data models is their dependence on the availability of comprehensive and representative training data. In the field of maintenance, data on failures are of particular interest, but these are often not available because failures are prevented or rarely occur. Additionally, data models may struggle with interpreting highly complex or noisy data, leading to suboptimal performance in certain scenarios. In this paper, the authors propose to combine knowledge and data models to obtain better predictions in a model-based CM framework.

The structure of the paper is as follows: First, more details on CBM and CM are given with an emphasis on model-based approaches in Section 2. This section also presents background on modeling, simulation,

979-8-3315-3420-2/24/$31.00 ©2024 

and modeling in simulation. Furthermore, basics of ML as well as RNNs in particular are introduced. The section ends with an overview of existing approaches that combine simulation and ML. In Section 3, the novel approach to combine simulation and ML is introduced and discussed. Subsequently, the approach is demonstrated in Section 4 on an industrial use case of a vacuum process in an industrial furnace. The paper closes with a summary and further research opportunities.

## 2 RELATED WORK

This section introduces the related work and its background. First, an overview of the application domain of CBM and CM is given. Then, modeling and the simulation of models are presented, as they are an important part of model-based CM. Other types of models and how they can be created from data are addressed in the context of ML and RNNs. Finally, a short overview of existing approaches to combine simulation and ML is given.

### 2.1 Condition-based Maintenance and Condition Monitoring

Maintenance is an ongoing process that encompasses all technical, administrative, and managerial actions throughout a machine's lifespan aimed at preserving or restoring its functionality to meet operational needs (DIN 2018). The deviation of at least one functionality of a machine from an acceptable state is referred to as a fault. The overall state of a machine with regard to the extent to which it is affected by faults is called the condition, whereby a good condition means that no faults are present and a poor condition means the presence of faults. To allow gradations in the condition regarding the severity of faults, there are not only the discrete values good and bad, but other suitable discrete or continuous scales are possible as well.

Maintenance is carried out using one of three strategies (Jardine et al. 2006). The first strategy is called reactive or breakdown maintenance, where maintenance measures are only conducted after faults have happened. The second strategy is preventive maintenance, where maintenance is conducted in periodic time intervals, regardless of the machine's condition. The most promising strategy is referred to as CBM. Here, maintenance measures are conducted when the condition of the machine indicates their necessity. The goal is to predict the occurrence of faults in advance, in order to plan measures beforehand, which prevents unexpected downtime, reduces costs, and enhances efficiency (Isermann 2011). To emphasize the prediction aspect of CBM, it is also often referred to as predictive maintenance.

To enable CBM, the condition of a machine must be constantly monitored and assessed. CM is based on data collected by sensors during machine operation. As those data are observed from reality, they are referred to as observational data and are prone to flaws regarding their data quality. Flaws include, e.g., measurement inaccuracies, missing data, and outliers (García et al. 2015). Appropriate data pre-processing measures should be taken to ensure sufficient quality for subsequent processes. The two main tasks in CBM are fault detection and fault diagnosis on the observational data (Isermann 2011). Fault detection uncovers and quantifies deviations from the expected behavior. If there is any deviation found by fault detection, it is further analyzed regarding its cause and magnitude in fault diagnosis.

An established approach for fault detection is to compare the measured observational data to a model of the monitored process. The model can represent the machine in either a good condition or a condition that includes degradation. A difference between the observed data and the model indicates a deviation in the expected behavior. The difference is referred to as the output error or residual. The output error is further assessed in the fault diagnosis by, e.g., statistical tests and threshold checking (Ding 2013). The fault diagnosis of residuals is also referred to as residual evaluation.

### 2.2 Modeling and Simulation

Models are a simplified replica of a planned or real system with its processes (VDI-Guideline 3633 Part 1 2014). They are never identical to the system to replicate and achieve different degrees of abstraction, which depend on the user's objective. Therefore, modeling is a subjective process and, thus, for the same modeling task, a multitude of suitable model solutions exist. There are different types of models, e.g., taxonomies or process models. In this paper, the focus is on mathematical, simulation, and ML models. Industrial companies use these to predict the behavior of systems or machines, which has proven to be value-adding (Dubois 2018). A limitation of modeling is its inherent inaccuracy due to abstraction, which is one of the main challenges of model-based CM to overcome (Badihi et al. 2022).

Simulation is a method to analyze systems and their dynamic processes with experiments on models (Law 2015). Those models are referred to as simulation models and are usually processed over time leveraging simulation tools. Simulation is used in particular for complex systems where analytical methods reach their limits. Modeling for simulation is mainly based on a-priori knowledge, making them knowledge models. The process to create models for simulation should be guided by a procedure model. An established procedure model for simulation modeling is proposed by Rabe et al. (2008), which consists of three phases. Given a task description, the first phase is the system analysis, where a conceptual model is derived. In the second phase, the conceptual model is formalized. The last phase is the implementation for the used simulation tool, which implements the formal model as an executable simulation model. An important aspect in modeling, including simulation modeling, is verification and validation (V&V) to ensure that the model is constructed correctly (verification) and adequately represents the real-world system it intends to replicate (validation) (Rabe et al. 2008).

### 2.3 Machine Learning and Recurrent Neural Networks

ML is an established and popular field of approaches to build models from data, which is also referred to as learning or training. Contrary to knowledge models, they are referred to as data models. In particular, this paper focuses on ML by supervised learning, where training runs on labeled data, and regression, which is the prediction of values based on previously learned data patterns, with each characteristic in a pattern termed feature (James et al. 2023). Popular methods for implementing regression with supervised learning are artificial neural networks (ANNs).

ANNs have achieved remarkable results in pattern recognition in, e.g., text, speech, and data (Goodfellow et al. 2016). They consist of neurons that output real values based on inputs, which are either values provided from outside the ANN or the weighted output of other neurons in the ANN, biases, and an activation function that transforms the inputs. Nowadays, the rectifier linear unit (ReLU) is the most widely used activation function (Zhou 2021). Neurons in ANNs are organized in layers, with the number of parallel neurons referred to as the number of units. Layers are categorized as input, output, and hidden layers. ANNs always have one input layer, representing external values to be processed, and one output layer, producing the ANN's output. Hidden layers, situated between the input and output layers, connect only to other neurons in the ANN. The number of hidden layers varies based on the problem. Often, neurons in one layer receive inputs from all neurons in the previous layer, forming a dense layer. The ANN's overall structure, including input and output shapes, number of layers, units, and activation functions, is its architecture. Determining the best architecture requires structured experimentation (Zhou 2021).

Given an architecture of one hidden layer, the ANN is only capable of processing the input data directly. However, this representation has proven to be difficult for computers. Empiric studies have shown that adding more layers to the architecture helps the algorithm to transform data to more beneficial representations between layers, which results in a better generalization for many tasks (Goodfellow et al. 2016). As multiple layers enhance the depth of an architecture, ANNs with multiple layers are referred to as deep ANNs. There is a branch of ML dedicated to such deep ANNs that is called deep learning, which is designed to learn on problems in high-dimensional spaces. A comprehensive guide on deep learning is given by Goodfellow et al. (2016).

Neural networks can be categorized according to the flow of data inside. If data flow through all layers sequentially, regardless of previous iterations (except training), these are referred to as feedforward or acyclic networks. This type of networks is suitable for static input-output mappings. When previous iterations should also influence following ones by maintaining an internal state, feedback is included and the

ANN works cyclic. Those ANNs are then referred to as RNNs. They are suited to process sequential data of variable length, such as text and time series data. The currently best-performing types of architecture for RNNs in a wide variety of tasks are the long short-term memory (LSTM) and gated recurrent unit (GRU) architectures (Schmidhuber 2015). Detailed explanations on RNNs can be found in Graves (2013). Another type of ANNs proposed by (Vaswani et al. 2017) are Transformers, which gained lots of attention and achieve better solutions in many tasks of natural language processing, computer vision, and time series prediction in general. However, tasks where accurately modeling the order of temporal changes is crucial can be considered a weak area of application for Transformers due to the tasks' permutation-invariant nature, and RNNs arguably perform better while retaining small model sizes (Zeng et al. 2023).

Training an ANN means adjusting the weights and biases in such a way that a desired behavior is achieved (Schmidhuber 2015). The training happens iteratively on training dataset in epochs. The performance of an ANN in training is measured by metrics like the mean squared error or the mean absolute error (MAE) (Zielesny 2016). The optimization of the weights and biases is based on minimizing a loss function that uses one of the previously mentioned metrics and is often realized by iterative, gradient-based methods such as adaptive moment estimation (commonly known as Adam), which also feature adaptive learning rates (Goodfellow et al. 2016). To enhance the training process, additional validation data are utilized. This is a separate set of data, which is used to test the ANN's performance on unseen data.

A challenge of ML models is that they usually do not follow any known physical laws, but are based purely on data (Jardine et al. 2006). The integration of algebraic or differential equations into the learning process, e.g., by using loss functions that incorporate them, is referred to as informed ML (von Rueden et al. 2021). Specifically for ANNs, the term physics-informed neural network has been established.

### 2.4 Combining Simulation and Machine Learning

Simulation and ML are well-established and widely-used methods that share a similar goal to predict system behavior and can also be integrated. A taxonomy of informed ML, which includes the integration of simulation, is reported by von Rueden et al. (2021). In previous research, von Rueden et al. (2020) identified three ways to combine simulation and ML. First, there is simulation-assisted ML, where simulation adds input to the ML, e.g., by generating synthetic training data, or is used to validate results. An example is reported by Rabe et al. (2017), where reinforcement learning is used to learn a policy for improving a logistics network with the help of a simulation model. Second, ML-assisted simulation can be used, where ML, e.g., creates surrogate models of parts that are hard to simulate. Also, a third - but rarely utilized - way is to use a hybrid approach, where simulation and ML are closely entangled. The combination of simulation and RNNs in particular has also been proposed in the past. Zhang et al. (2019) demonstrated the successful integration of a simulation model of bearing performance degradation, capable of generating an entropy indicator, to generate training data for a RNN employing a LSTM architecture.

There is an increasing amount of publications in this area and von Rueden et al. (2020) emphasize the great potential of combining knowledge models with data models for applications that are partly based on causal relationships as well as hidden dependencies in data, especially in the domain of the industrial digital transformation. The great potential of combining simulation and ML is also recently emphasized by a position paper of leading members of the simulation community reported by Taylor et al. (2023), especially in regards to applications within digital twins.

With the approach presented in this paper, the authors contribute to the research field of combining simulation and ML in the domain of CBM. The approach's novelty consists of the integration of a simulation model and a RNN not to train the RNN by simulation, like done by Zhang et al. (2019), but to use it as a combined model in the operational phase. With this approach, the challenges of insufficient (high-quality) data for ML and the inaccuracies in simulation model creation, as highlighted in the literature, can be addressed. These aspects will be further elaborated in the following sections.

## 3 APPROACH TO COMBINE SIMULATION AND RECURRENT NEURAL NETWORKS FOR MODEL-BASED CONDITION MONITORING

This section introduces the novel approach by first discussing the general idea behind it. This idea is then explained in detail using a schematic process and a procedure model in Section 3.2.

### 3.1 General Idea

For model-based CM, a high-quality model of the machine to monitor is essential. High quality in this context refers to a model that accurately represents the behavior, dynamics, and intricacies of the machine under various operating conditions. This entails capturing both the regular operation of the machine and the deviations that may indicate potential faults or failures. The model should be sufficiently detailed to encompass the key components, interactions, and environmental factors affecting the machine's performance. Additionally, the model should be adaptable and updatable to accommodate changes in the machine's configuration or operating conditions over time. Furthermore, it should be robust enough to handle uncertainties and variations inherent in real-world operations. Overall, a high-quality model for model-based CM serves as a reliable foundation for effective fault detection and diagnosis, ultimately facilitating timely maintenance and minimizing downtime.

As explained in Section 2.2, modeling is a process that is subject to uncertainty and assumptions. This includes the modeling of physical processes. For example, gases are often considered as ideal gases under the assumptions of constant temperature and pressure, negligible volume of gas particles, and non-interacting particles. Although ideal gas is a good model for many applications, it is still imperfect. To make the modeling of gases more realistic, comparatively complex analytical processes or simulations are used, which can increase the effort considerably. However, even then, the model will be, though less, imperfect. A common aphorism in the modeling community (often attributed to George Box, e.g., in Box 1979, p. 202) is "all models are wrong but some are useful". With regard to this popular quote, the idea of this paper is to create models that are less wrong and, therefore, more useful.

To achieve this, the authors propose to combine a-priori knowledge, in the form of simulation models, with data patterns, incorporated in data models created by ML. The general idea is to leverage available knowledge for a well-founded basis and use this data model to adjust it to previously seen observational data. Combining a simulation model with an ML data model is advantageous when insufficient data, including run-to-failure data, are available. The combined model can rely on simulation predictions, ensuring good predictions without observational data. This approach is also beneficial for constantly new scenarios lacking sufficient data, even if some data are available for other cases.

A challenge in model-based CM is that the used knowledge models often replicate an idealized behavior of a machine. However, the sensors used to measure the observational data are subject to measurement inaccuracies, which will not result in an idealized behavior. For this reason, there will almost always be deviations. This situation is illustrated on an example in Figure 1.

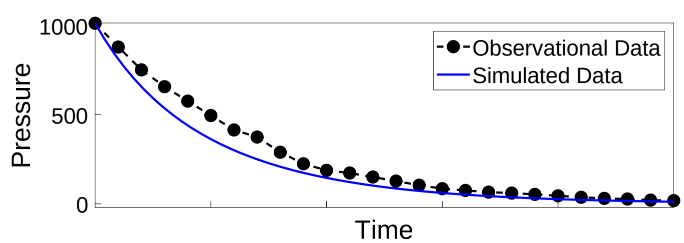


Figure 1: Example of simulated data compared to observational data.

Figure 1 shows observational data of a vacuum process (reduction of pressure in a closed environment) as well as corresponding simulated data. The simulated data series resembles the idealized physical laws of a vacuum process. Contrary, the observational data are not as smooth as the simulated ones and also have slightly different values for each point in time. These inaccuracies encompass both systematic measurement errors, such as calibration flaws and limitations in resolution, and random measurement errors, like environmental noise. To ensure that no deviations are measured for the case shown, i.e., that the residuals have the value zero, the simulation model would have to be adapted so that it calculates the observational sensor values rather than the ideal values. This adaption is nearly impossible to realize by a human modeler, as the underlying measurement inaccuracies are hardly known. However, as described in Section 2.3, ML is suited for this task and can learn from the data directly. By using the simulation results, which represent a good approximation of the behavior, as a basis, the number of features to be learned is reduced by ML.

The problem at hand can be categorized into the class of supervised learning methods, as the data to learn from are labeled. Specifically, the task is to create a regression model for time series data. It is important to look at the data as sequences rather than in isolation, because sequences of data points exhibit temporal dependencies that can provide valuable information for prediction. This approach allows the model to learn from the past observations and make informed predictions about future values based on the historical context embedded within the time series. RNNs, which are described in Section 2.3, are an established method and particularly suitable for the aforementioned requirements.

In summary, the approach represents a succession of a simulation model and a RNN to increase the prediction quality. From a formal point of view, the model-based prediction for the residual generation can be stated as shown in the iterative Equation (1), where $t$ is a point in time, $y_t$ is the predicted value of the approach for $t$, $S$ is the simulation model, $\Theta$ the process input parameters, and $D$ the recurrent data model.

$$y_t = S(t,\Theta) + D(t,\Theta,S(t,\Theta),D(t-1,\Theta,S(t-1,\Theta))) \tag{1}$$

The output of the simulation model that is fed into the RNN must be a single predicted value for a given $t$. For simulation models with (high) stochastic influences, the usage of an appropriate design of experiments and a final averaging to determine the predicted value can be considered. As the simulation of physical laws often does not include stochastic characteristics, the usage of a design of experiments can be omitted.

The process input parameters $\Theta$ of Equation (1) are used to describe the machine's phyisical process that is currently predicted. Examples for such parameters are, e.g., available power for motors, the intended target temperature of heating processes, and information on the material that is currently being processed.

In the following section, a structured methodology is outlined to implement this approach.

### 3.2 Details of the Approach

In the following, the approach will be introduced in detail based on a procedure model. The approach can be used for measured variables that can be captured by sensors at the machine and that can be described sufficiently well using simulation models. Before selecting a measured variable for the approach, it should be ensured that a sufficiently large influence on the condition of the machine is assumed.

The approach is divided into two phases. The first phase is the model building phase, where the model for the model-based CM is built. The second phase is the operation, where the built model is utilized for CM.

The first phase is depicted as a procedure model in Figure 2. This phase is subdivided into six steps:

- **Step 1**: The first step is the simulation model building process. As shown in Section 2.2., the model building should be conducted according to a structured procedure model. Notably, the method outlined in this paper does not mandate a particular procedure model, affording users the flexibility to choose one that suits their needs. The simulation model should be able to output values for any given point of time where observations are possible. Therefore, continuous time simulation models are preferred, although discrete event simulation models can be used if values for all observation points can be derived through other means like, e.g., interpolation.

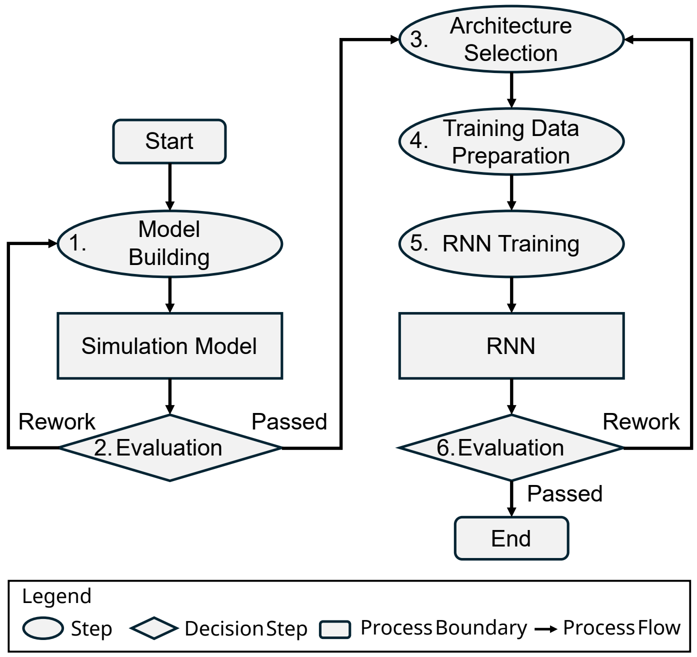


Figure 2: Model building phase procedure model.

- **Step 2**: Once a simulation model is constructed, it must undergo an evaluation through V&V procedures. If any issues are identified or if the results are unsatisfactory, it may be necessary to iterate through Step 1 multiple times until satisfactory outcomes are achieved. Many model building procedure models already incorporate this iterative loop, rendering it potentially redundant. Upon obtaining a positively evaluated simulation model, the user can advance to Step 3.
- **Step 3**: This steps marks the beginning of building an RNN. At first, an architecture must be selected. As explained in Section 2.3, the architecture of RNNs mainly depends on the selection of a number of layers, their type, and the number of units in it. The architecture should be selected to ensure that the RNN can accurately predict the deviation values.
- **Step 4**: In this step, a set of data must be prepared that will be used for training. It must comprise enough volume of heterogeneous data collected when the machine is in good health, representing various operational scenarios and conditions. This dataset ensures that the ML model can generalize well and effectively learn the patterns associated with regular operation.
- **Step 5**: Now, an RNN with the selected architecture of Step 3 must be trained on the training dataset of Step 4. This step uses the simulation model that was created in the first steps, as the simulation model's output is one of the inputs to the RNN (see Equation (1)). Information on successful training are given in Section 2.3.
- **Step 6**: This step assesses the suitability of the generated RNN. The evaluation should consider the training performance, focus on the loss function, and also involve visually testing exemplary predictions. If the RNN's performance is unsatisfactory, Steps 3 to 6 can be iterated. Upon achieving a satisfactory RNN, the procedure model concludes, marking the end of the first phase.

With both a knowledge and data model available, the operation phase can commence. Upon combining these models, they effectively merge into a unified entity, representing a single comprehensive model. A schematic overview for the second phase is shown in Figure 3.

In the second phase, the procedure mostly aligns with a standard implementation of model-based CM. A physical machine in a real-world setting receives process input, encompassing instructions on its intended

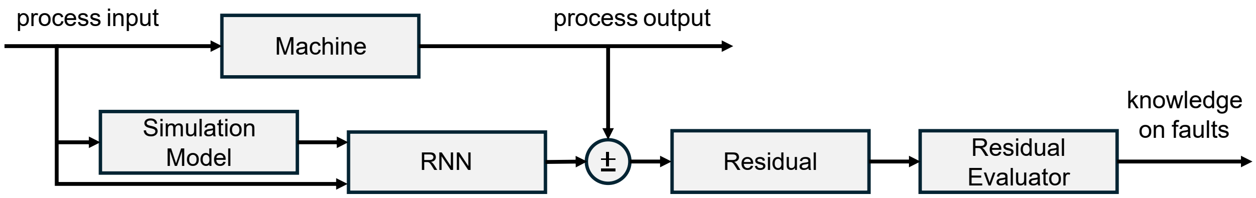


Figure 3: Operational phase schematic.

actions along with the specified parameters and conditions. The machine subsequently operates based on this process input, generating a process output. This output includes information on the final outcome, alongside data accumulated during the processing phase. Running in parallel to the physical machine, the same process input (corresponding to $\Theta$ in Equation (1)) is also fed into the simulation model. Utilizing its inherent knowledge, the simulation model computes an expected value. Both the output of the simulation model and the original process input are combined to form the input for the RNN. The RNN adjusts the anticipated value based on learned data patterns. Subsequently, the discrepancy between the RNN output and the machine's process output is computed. As outlined in Section 2.1, this disparity is termed the residual, which is then assessed using appropriate metrics to provide insights into faults, enabling CBM (see Section 2.1).

The introduced novel approach is demonstrated using an industrial use case in the next section.

## 4 USE-CASE-BASED PROOF OF CONCEPT

In this section, an exemplary CM on previously recorded observational data is shown. The data contain measurements for levels of pressure in the heating chamber of an industrial furnace during vacuum processes. The industrial furnace is mainly used for heat treatment of metals and requires a vacuum to allow for a controlled atmosphere and prevent reactions with oxygen in the air. The vacuum is created by a vacuum pump and the sensor used is a vacuum transducer.

The modeling of a vacuum process in a heating chamber can be described by two terms: on the one hand by the work of the pump, which reduces the pressure, and on the other hand by a leakage rate, which increases the pressure as air flows in through the enclosure or piping of the heating chamber. The work of the pump is dependent on the pump speed $s$ [$\frac{\mathrm{m}^3}{\mathrm{s}}$] and the volume of the heating chamber $V$ [$\mathrm{m}^3$]. The leakage rate $l$ [$\frac{\mathrm{mbar}}{\mathrm{s}}$] is dependent on the difference of pressure outside and inside the heating chamber described as $p_{at} - p$, where the standard atmosphere pressure is defined as $p_{at} = 1013.25$ mbar. From this, a mathematical formulation by a differential equation can be derived for the change of pressure $p$ [mbar] in the heating chamber that is given in Equation (2).

$$\frac{\partial p}{\partial t} = -p\frac{s}{V} + l\sqrt{\frac{p_{at} - p}{\mathrm{mbar}}} \tag{2}$$

To use Equation (2) for time-continuous simulation, an incremental version must be derived. With time steps $\Delta t_i$ that must be sufficiently small, an incremental version can be obtained by utilizing the Euler method, which results in Equation (3).

$$p_i = p_{i-1}e^{\left(-\frac{s}{V}\Delta t_i\right)} + l\sqrt{1013.25 - p_{i-1}}\Delta t_i \tag{3}$$

The parameters of Equation (3) correspond to the process input parameters $\Theta$ in Equation (1). As Equation (3) is derived from well-known physical laws and was already tested by (Wuttke et al. 2023), the evaluation to employ it, as outlined in Step 2 of the phase procedure model, is affirmative.

The next step is to choose an architecture for an RNN to describe the modeling uncertainty. As described in Section 2.3, an architecture must be found by experiments. The authors experimented with several common RNN architectures with the layer types dense, LSTM, and GRU (see Section 2.3) with varying numbers

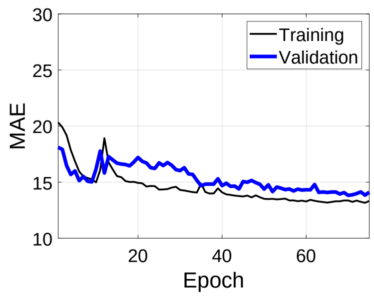


Figure 4: Combined model training.

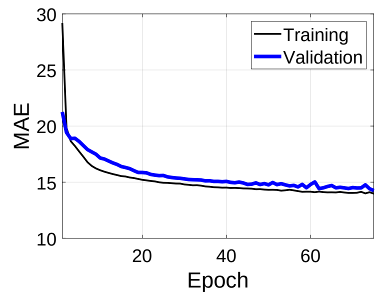


Figure 5: ML model training.

of layers and units. For this paper, one positively evaluated and well-performing architecture is presented. It uses one GRU and three dense layers with 64 or 32 neurons, which is a rather low number of units. Architectures with more layers and units were tested, but did not yield significantly better results. The details of the architecture can be found in Table 1. The training data are based on recorded observational

Table 1: RNN architecture for the use case.

| Layer | Type | AF | Units |
|---|---|---|---|
| Input | - | None | - |
| Hidden 1 | GRU | RelU | 64 |
| Hidden 2 | Dense | RelU | 64 |
| Hidden 3 | Dense | RelU | 32 |
| Hidden 4 | Dense | RelU | 32 |
| Output | - | None | 1 |

AF = activation function

data that were provided by an industrial partner. It is known from empirical observations during the time of recording that the provided data mainly include data on the machine in a good state, but also on a severe fault. Data collected when the machine was proven to be in good condition were used for the training and validation set. A set of 500 vacuum processes were utilized for the training dataset, while 250 vacuum processes were allocated for validation purposes. The process input parameters $\Theta$ were assumed to be constant. The training was conducted with the Adam optimizer (see Section 2.3) at a specified initial learning rate of 0.001 in 75 epochs. The performance of the RNN in training is visualized in Figure 4. To compare the RNN used in combination with simulation in the novel approach to an RNN with the same architecture used in a solely ML approach, a similar training was conducted, whose results can be seen in Figure 5. In the remainder of the section, the model of the novel approach is referred to as the combined model, while the model in the solely ML approach is referred to as the ML model.

From visual examination of the training results it can be seen that the performances of the models, measured with the MAE, converge, which is a sign of successful training. Moreover, the results show that the combined model has superior performance in the first epochs, which can be explained by the preexisting a-priori knowledge. Further findings are that the training process of the combined model is more exposed to fluctuations and the achieved results are slightly better than in the ML model. In addition to the training process shown, further iterations were carried out to confirm these observations.

The performance of the obtained models was visually assessed through exemplary predictions, showcased in Figure 6. For comparison purposes, a sole simulation model was also assessed. The results show that all models are capable of predicting vacuum processes and the combined model's predictions are very similar to the predictions of the simulation model. Based on the training performance and the initial assessment

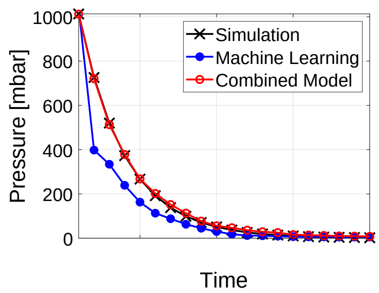


Figure 6: Exemplary predictions.

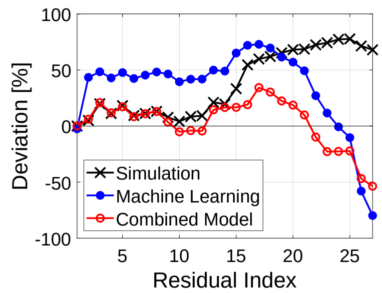


Figure 7: Deviations at Point 1.

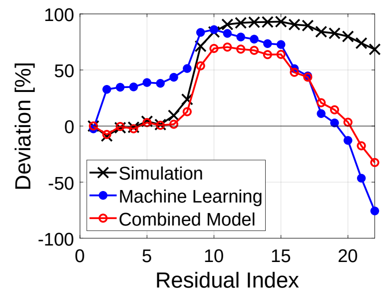


Figure 8: Deviations at Point 2.

of the prediction quality, the combined model is evaluated positively. This corresponds to Step 6 of the phase procedure model and marks the end of the first phase.

Phase two utilizes the remaining data from the dataset, which are 2.449 vacuum processes. The chosen metric for evaluating residuals is the relative deviation in percent. Alongside presenting the combined model's results, outcomes from utilizing the simulation model and the RNN independently are also provided for comparison purposes. The results of CM are reported for two points of time. "Point 1" refers to a point of time when the industrial furnace was known to be in good condition, whereas "Point 2" denotes a point of time when the furnace was known to be in bad condition.

The results of Point 1 are visualized in Figure 7. The best performing model is the combined model, whose average deviation is the lowest. Its pattern is similar to the data of the ML model, however, the deviations are about half the size. In the first half of the process, the combined model nearly matches the simulation values, but feature significantly better results in the second half. In Figure 8, the results of Point 2 are shown. The average deviation is also lowest in the combined model, which means it has the highest prediction quality. Compared to Point 1, the average deviation of all three models has increased, which means the state of operation in Point 2 differs more from the nominal than in Point 1. As all three models detected this change, they are all suitable models for model-based CM.

For the presented experiment, the combined model has proven to yield superior results compared to the sole use of ML and simulation. However, in experiments with more complex architectures of RNNs not reported in detail in this paper, the discrepancy to the sole use of ML is reduced although the combined model always maintained at least slightly better results.

Up to this point, the CM was conducted on the same set of parameters $\Theta$. To investigate the transferability performance on other parameters, a second scenario was assessed, where the volume $V$ was set to 10 % of the original value. The results are visualized in Figure 9 and show an advantage of the combined model over the ML model regarding the prediction quality on new scenarios. As the training was only conducted for the original value of $V$, the ML model is not able to predict the new scenario well. However, process parameters change frequently in industrial practice. Therefore, the novel approach demonstrates particular effectiveness in scenarios where a-priori knowledge can be leveraged, such as frequent encounters with new situations or insufficient training data to adequately train a sole ML model.

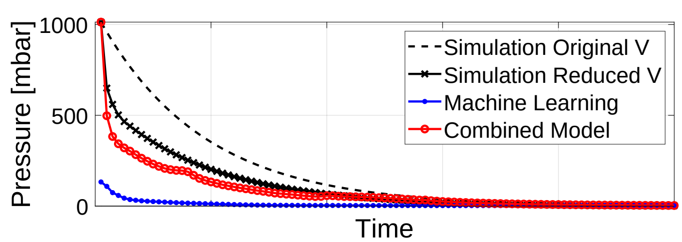


Figure 9: Prediction in the new scenario.

## 5 SUMMARY AND OUTLOOK

This paper introduces a novel approach that combines simulation and ML, particularly RNNs, within the domain of model-based CM to facilitate CBM of machines. After providing the theoretical background and reviewing related work, the paper outlines the fundamental concept behind the approach. It acknowledges that simulated data often predict an idealized trajectory that does not align with observed data from sensors due to measurement and modeling inaccuracies. Also, pure ML approaches reach their limit when insufficient data are encountered. To address these problems, the paper proposes a novel approach with two phases: the first involves constructing a combined model comprising a simulation model and an RNN, while the second entails the operational phase of the combined model within a model-based CM process. The core idea of the combined model is to use simulation first, to predict a first good prediction by a knowledge model, and then feed this into an RNN, to adjust the prediction by a data model to overcome modeling shortcomings. The effectiveness of the novel approach is tested using an industrial use case involving vacuum processes in an industrial furnace, leveraging a pre-existing dataset. A comparison is drawn between the combined model and the sole use of the simulation model and RNN. The results demonstrate superior fault identification capabilities of the combined model as well as its effectiveness on predicting previously unseen scenarios. If sufficient data for all possible scenarios are available or a suitable simulation model exists, it may be unnecessary to invest additional effort in creating a combined model. Otherwise, a combined model can provide comprehensive insights and improve overall accuracy.

Future research should explore several aspects of the novel approach. Firstly, it should be applied to other use cases involving different processes and machines to assess its versatility. Secondly, enhancing residual evaluation by employing additional metrics could lead to improved fault identification and diagnosis. Also, the usage of Transformers over RNNs can be a promising approach. Additionally, deploying the approach within the operational setting of the industrial domain would provide further insights into its performance. Furthermore, integrating the approach into a digital twin framework, which is a popular area of research aimed at enhancing CBM, holds promise for future investigations.

## REFERENCES

Badihi, H., Y. Zhang, B. Jiang, P. Pillay and S. Rakheja. 2022. "A Comprehensive Review on Signal-Based and Model-Based Condition Monitoring of Wind Turbines: Fault Diagnosis and Lifetime Prognosis". *Proceedings of the Institute of Electrical and Electronics Engineers, Inc.* 110(6):754–806.

Box, G. 1979. "Robustness in the Strategy of Scientific Model Building". In *Robustness in Statistics*, edited by R. L. Launer and G. N. Wilkinson, 201–236. Amsterdam: Elsevier.

DIN 2018. *DIN EN 13306:2018-02, Maintenance Terminology; Trilingual Version*. Berlin: Beuth.

Ding, S. X. 2013. *Model-Based Fault Diagnosis Techniques*. London: Springer London.

Dubois, G. 2018. *Modeling and Simulation: Challenges and Best Practices for Industry*. Boca Raton, FL: CRC Press.

García, S., J. Luengo, and F. Herrera. 2015. *Data Preprocessing in Data Mining*. Cham: Springer International Publishing.

Goodfellow, I., Y. Bengio, and A. Courville. 2016. *Deep Learning*. Cambridge, MA: MIT Press.

Graves, A. 2013. *Generating Sequences With Recurrent Neural Networks*. arXiv https://doi.org/10.48550/arXiv.1308.0850.

Isermann, R. 2011. *Fault Diagnosis of Technical Processes, Applications: Model Based Methods for Actuators, Sensors, Drives, Machinery, and Industrial Plants*. Berlin: Springer.

James, G., D. Witten, T. Hastie, R. Tibshirani and J. Taylor. 2023. *An Introduction to Statistical Learning*. Cham: Springer International Publishing.

Jardine, A. K., D. Lin, and D. Banjevic. 2006. "A Review on Machinery Diagnostics and Prognostics Implementing Condition-based Maintenance". *Mechanical Systems and Signal Processing* 20(7):1483–1510.

Law, A. 2015. *Simulation Modeling and Analysis*. 5th ed. New York City, NY: McGraw-Hill Education.

Quatrini, E., F. Costantino, G. Di Gravio, and R. Patriarca. 2020. "Condition-Based Maintenance: An Extensive Literature Review". *Machines* 8(2):31.

Rabe, M., F. Dross, and A. Wuttke. 2017. "Combining a Discrete-event Simulation Model of a Logistics Network with Deep Reinforcement Learning". In *Proceedings of the MIC and MAEB 2017 Conferences*. July 4$^{th}$-7$^{th}$, Barcelona, Spain, 765-774.

Rabe, M., S. Spieckermann, and S. Wenzel. 2008. "A New Procedure Model for Verification and Validation in Production and Logistics Simulation". In *2008 Winter Simulation Conference (WSC)*, 1717–1726 https://doi.org/10.1109/WSC.2008.4736258.

Schmidhuber, J. 2015. "Deep Learning in Neural Networks: An Overview". *Neural Networks : The Official Journal of the International Neural Network Society* 61:85–117.

Taylor, S. J. E., C. M. Macal, A. Matta, M. Rabe, S. M. Sanchez and G. Shao. 2023. "Enhancing Digital Twins with Advances in Simulation and Artificial Intelligence: Opportunities and Challenges". In *2023 Winter Simulation Conference (WSC)*, 3296–3310 https://doi.org/10.1109/WSC60868.2023.10408011.

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, *et al*. 2017. "Attention Is All You Need". In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS)*. December $4^{\text{th}}$-$9^{\text{th}}$, Long Beach, CA, USA, 5998-6009.

VDI-Guideline 3633 Part 1 2014. *VDI 3633 – Simulation of Systems in Materials Handling, Logistics and Production: Fundamentals*. Berlin: Beuth.

von Rueden, L., S. Mayer, K. Beckh, B. Georgiev, S. Giesselbach, R. Heese *et al*. 2021. "Informed Machine Learning: A Taxonomy and Survey of Integrating Prior Knowledge into Learning Systems". *IEEE Transactions on Knowledge and Data Engineering*:1.

von Rueden, L., S. Mayer, R. Sifa, C. Bauckhage and J. Garcke. 2020. "Combining Machine Learning and Simulation to a Hybrid Modelling Approach: Current and Future Directions". In *Advances in Intelligent Data Analysis XVIII*, edited by M. R. Berthold, A. Feelders, and G. Krempl, 548–560. Cham: Springer International Publishing.

Wuttke, A., J. Hunker, M. Rabe, and J.-P. Diepenbrock. 2023. "Estimating Parameters with Data Farming for Condition-based Maintenance in a Digital Twin". In *2023 Winter Simulation Conference (WSC)*, 1641–1652 https://doi.org/10.1109/WSC60868.2023.10408594.

Zeng, A., M. Chen, L. Zhang, and Q. Xu. 2023. "Are Transformers Effective for Time Series Forecasting?". In *Proceedings of the 37th AAAI Conference on Artificial Intelligence*. February $7^{\text{th}}$-$14^{\text{th}}$, Washington, DC, USA.

Zhang, B., S. Zhang, and W. Li. 2019. "Bearing Performance Degradation Assessment Using Long Short-term Memory Recurrent Network". *Computers in Industry* 106:14–29.

Zhou, Z.-H. 2021. *Machine Learning*. Singapore: Springer Singapore.

Zielesny, A. 2016. *From Curve Fitting to Machine Learning*. Cham: Springer International Publishing.

## AUTHOR BIOGRAPHIES

**ALEXANDER WUTTKE** is a researcher at the department IT in Production and Logistics at TU Dortmund University. He holds a Master of Science in Mechanical Engineering with a focus on IT in Production and Logistics from TU Dortmund University. His research interests are digital twins, condition-based maintenance, and simulation. His email address is alexander2.wuttke@tu-dortmund.de.

**MARKUS RABE** is a full professor for IT in Production and Logistics at the TU Dortmund University. Until 2010 he had been with Fraunhofer IPK in Berlin as head of the corporate logistics and processes department, head of the central IT department, and a member of the institute direction circle. His research focus is on information systems for supply chains, production planning, and simulation. Markus Rabe is vice chair of the "Simulation in Production and Logistics" group of the simulation society ASIM, member of the editorial board of the Journal of Simulation, member of several conference program committees, has chaired the ASIM SPL conference in 1998, 2000, 2004, 2008, and 2015, Local Chair of the WSC'2012 in Berlin and Proceedings Chair of the WSC'2018 and WSC'2019. More than 250 publications and editions report from his work. His email address is markus.rabe@tu-dortmund.de.

**JOACHIM HUNKER** is a researcher and the deputy head of department industrial manufacturing at Fraunhofer Institute for Software and Systems Engineering (ISST). He holds a Master of Science in Logistics, Infrastructure, and Mobility with a focus on IT in Logistics from the Technical University of Hamburg. He graduated with a master thesis on a hybrid scheduling approach of assembly lines of car manufacturers. His research focuses on simulation-based data generation and data analytics in logistics. His email address is joachim.hunker@tu-dortmund.de.

**JAN-PHILIPP DIEPENBROCK** is an engineer at IVA Schmetz GmbH, Germany. He holds a Ph.D. from the University of Magdeburg and researched in the field of simulation and data processing. He also holds a diploma in Physics from TU Munich University and studied at the University of Cambridge under a scholarship. His research focuses on machine learning for condition monitoring and condition-based maintenance. His email address is jan.diepenbrock@iva-schmetz.de.