Power Consumption Forecasting for predictive maintenance of a Linear Axis Motion System using Deep Learning algorithms

TUM School of Management, WS 2022/2023

Data Analytics in Applications
Prof. Dr. Dr. h. c. mult. Horst Wildemann
Sebastian Junker
Niklas Stepanek

In Cooperation with Scheider Electric
Dr. Sebastian Furth
Pankaj Kumar Shaw

Fober, Luca
luca.fober@tum.de

Sammer, Maximilian
maximilian.sammer@tum.de

Siebenbürger, Mathias
mathias.siebenbuerger@tum.de

***Abstract*— Schneider Electric has taken up the challenge to optimize the functionality of their industrial machines through the prediction of the power consumption for better production rates and quality. Taking the CRISP-DM model into account, we developed a Deep Learning algorithm with a multiple input and multiple output strategy for the future prediction. The Deep Learning model consists of an encoder-decoder architecture by stacking RNN layers. We used this architecture to test the performance of two variants of RNN, namely GRU and LSTM, with similar outcomes. With this model, we can predict the machine power consumption – despite having some limitations – with good accuracy and therefore are able to deliver the base for a predictive maintenance solution at Schneider Electric.**

Keywords—power consumption forecasting, Deep Learning, time series prediction, Predictive Maintenance

I. Introduction and Basics

One of the global megatrends is digitalization. In particular the Corona pandemic has massively increased the importance of digitization for companies (Streim and Meinecke 2020). A study of Bitkom, the Industry association of the German information and telecommunications sector, shows the increasing potential of Internet of Things platform-based applications within Industry 4.0 (Berg 2020, Manyika et al. 2015). One of those promising applications is predictive maintenance (Zonta et al., 2020; Lorenz et al., 2016; Sirkin et al., 2015).

A. Definition and Potential of Predictive Maintenance

Predictive maintenance is a proactive maintenance strategy that utilizes data analytics and machine learning techniques to predict the likelihood of equipment failure or the need for maintenance. This approach allows organizations to schedule maintenance and repairs before equipment failure occurs, thereby reducing downtime and improving overall equipment efficiency (Cortes et al., 2021; Pech et al., 2021; Olesen & Shaker, 2020; Dennis et al., 2017; Ran et al., 2019; Schreiner and Mundt, 2020; Waterer, 2012). Predictive Maintenance has the potential to revolutionize the way in which organizations approach equipment maintenance, when comparing it to Reactive or Preventive maintenance (Zonta et al., 2020; Paquin, 2014). Reactive maintenance is characterized by repair work conducted after a failure or breakdown, often completed as an emergency (Ran et al., 2019). Preventive Maintenance describes routinely performed maintenance, which does not take the condition of the equipment into account, and therefore may be inefficient (Aksa et al., 2021; Elahi et al., 2022; Ran et al., 2019; Selcuk, 2015; Waterer 2012). In contradiction to these two maintenance strategies, Predictive Maintenance allows a company to proactively manage their equipment and improve efficiency as well as profitability, as Predictive Maintenance is the least expansive maintenance strategy (Ran et al., 2019; Paquin, 2014; Waterer, 2012).

The use of sensors to monitor equipment performance and the analysis of data on past failures enables organizations to identify patterns and predict when equipment is likely to fail (Fraunhofer IPA, 2019). This enables the scheduling of maintenance at a convenient time, rather than reacting to equipment failures as they occur. Additionally, Predictive Maintenance can help organizations optimize their maintenance schedules and reduce the need for unnecessary repairs by identifying the root cause of equipment failures and implementing preventative measures to extend the lifetime of the equipment (Cortes et al., 2021; Decaix et al., 2021; Paquin, 2014). Further specific applications of Predictive Maintenance include automatic tracking and the indication of when regular maintenance is due, optimization of the quality of manufactured products in a targeted manner, automatically detection of wear, optimization of setup processes, avoidance of faulty operation, detection of poor machining processes and display setup errors (Staufen, 2019). Predictive Maintenance can be applied in various asset-intensive industries, including chemicals, oil and gas, mining, metals, pulp and paper and power production (Dennis et al., 2017; Cortes et al., 2021; Decaix et al., 2021).

Despite the challenges of implementing Predictive Maintenance at scale, the potential benefits make it an important consideration for companies seeking to optimize their operations and increase profitability (Cortes et al., 2021; Dalzochioa et al., 2020; Olesen & Shaker, 2020). According to the consulting firms McKinsey & Company and Capgemini, predictive maintenance can generate substantial savings by increasing production line availability through breakdown elimination and downtime reduction by five to 15 %, while reducing maintenance costs by 18 to 25 % through for example the reduction of scheduled repairs (Bradbury et al., 2018; Barriball et al., 2017; Dennis et al., 2017). Moreover, reductions in the maintenance staff can be realized (Waterer 2012).

B. *Current Deploymentf Predictive Maintenance in Germany*

However, the successful implementation and use of Predictive Maintenance applications are currently not the industry standard in Germany. The survey by Staufen (2019), which involved 323 German representatives from the mechanical and plant engineering industries as well as from the electrical engineering and the automotive industry, shows that 38 % of the participants are not using any Predictive Maintenance solutions at all, while the previously described applications are also not widely spread in the industry (Staufen, 2019).

This need for action by German industry regarding Industry 4.0 is also described by Stich et al. (2019). According to Stich et al., the future vision of a smart factory can only be successful if maintenance develops into so-called Smart Maintenance. Predictive Maintenance is part of Smart Maintenance and is sometimes even used as a synonym (Fraunhofer IPA, 2019; Holst 2022).

C. *The use case of Schneider Electric*

An example of a company facing this digital transformation in maintenance is Schneider Electric, which is an electrical engineering group operating in the fields of electrical power distribution and industrial automation. Schneider Electric offers already various solutions that are capable of determining the operational status of the equipment, evaluating the present condition of the equipment and detecting abnormal conditions in a timely manner (Waterer, 2012).

In order to improve and expand the capabilities of Schneider Electric with respect to Predictive Maintenance, the company wants to develop a business case for the power consumption forecasting of industrial machines using a time series forecasting model based on a Deep Learning approach. According to Zhang et al. (2019), Deep Learning are delivering accurate results for Predictive Maintenance applications (Zhang et al., 2019).

D. *Definitions of Deep Learning and time series forcasting*

Deep learning is a subset of machine learning that involves training artificial neural networks on a large dataset. These neural networks are able to learn and make intelligent decisions on their own by analysing patterns in the data (LeChun et al., 2015).

Time series forecasting is the process of using historical data to predict future events. It involves analysing past trends and patterns in data to make informed predictions about future developments. Time series forecasting can be used in a wide range of applications, including financial analysis, demand forecasting and resource planning. It is often used in conjunction with statistical and machine learning techniques to build models that can accurately forecast future events (Hyndman and Athanasopoulos, 2018).

E. *Contribution and Organisation*

This paper provides a new state-of-the art model for the prediction of the power consumption of industrial machines, which can be used by the industry as an enabler for various predictive maintenance applications. Our model can detect abnormal power consumption patterns in industrial machines in real time. This enables the maintenance staff to both continuously monitor the condition of a certain machine and to initiate actions in case of abnormalities in order to prevent events such as machine failure or downtime. This improved condition monitoring of industrial machines can be translated into various benefits, such as the reduction of downtime or maintenance costs, resulting in increased profitability for a company.

We addressed this challenge based on the evaluation of a dataset from Schneider Electric, which included thousands of power consumption data points with respect to various speeds. The data points represent the power consumption of a developed motion system based on a linear axis with a single carrier operating on two different motion patterns: a discrete and a continuous motion pattern.

We structured the development of our time series forecasting model based on a Deep Learning approach according to the Cross Industry Standard Process for Data Mining (CRISP-DM) (Wirth and Hipp, 2000).

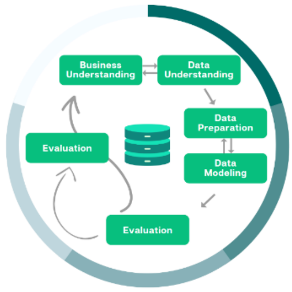


Figure 1: The cross-industry standard process for data mining (CRISP-DM) according to Wirth & Hipp 2000

Therefore, the remainder of this report is also structured according to the CRISP-DM model: First, we present our business understanding that underlies the challenge provided by Schneider Electric. Second, we will discuss our data understanding, before describing our data preparation in detail in the third chapter. Part four and five focus on the different Deep Learning models that we took into consideration. Consecutively, we evaluate the performance and the model accuracy. After briefly describing the deployment of our Deep Learning model, we will conclude by explaining the business implications for Schneider Electric, in chapter seven. Finally, we will provide key insights and identify the limitations of this study in chapter eight and summarize the results in chapter nine.

II. Business Understanding

For a data science project to be successful, it is essential to start with a deep understanding of the customer's situation and objectives. Hence, the first phase of business understanding focuses on the business context and requirements. Analysing and visualising the business of Schneider Electric using the Business Model Canvas by Osterwalder and Pigneur (2010) helped us to bring the scope and objectives of the project into one schematic overview in order to build up a business-related foundation of the problem (Appendix A). In the following paragraph, only the most important aspects of the canvas are highlighted.

A. *Business Model Canvas*

Downtime of industrial machinery is a significant challenge in several industries, including the automotive and processing industries. According to Schneider Electric, one hour of downtime causes costs of 2.5 million Euros in the automotive industry. In total, 24 % of total manufacturing costs are attributed to downtime (Schneider Electric, 2021). At the same time, most of the maintenance work is currently carried out in a reactive way instead of a preventive or predictive manner. Moreover, industrial machines are generating lots of data. Schneider Electric could generate useful insights into the current condition of a machine for their clients with the data gathered by their industrial automation and control products. In the majority industry cases, this potential remains unused even though the cost of reactive maintenance is five to 15 times higher than proactive maintenance (Schneider Electric, 2021).

B. Potential Business Case

Being able to provide clients with information about the current status and condition of their machinery is a very promising asset that could be translated into a competitive advantage. As Schneider Electric knows best how their machines are working, they could offer a subscription based Smart Maintenance Service that allows customers to anticipate possible machine failures in order to avoid production downtime. The condition of a machine could be analyzed using a Deep Learning algorithm for the prediction of future power consumption. In case of the detection of a deviating anomaly between the forecast and the actual data, an automatic recommendation will be sent to the operator on site. Henning et al. (2021) are describing how alerting in cases of anomaly detection could work in practice (Henning et al., 2021). These machine-operations experts can use this provision as a countermeasure in order to avoid machine downtime.

This business case expands Schneider Electric´s offerings towards a holistic product portfolio while filling the mentioned market gap in the transition from reactive maintenance to prognostic forecasts. In addition to this use case, there is a potential for further useful insights throughout Big Data analytics of machinery data. For Schneider Electric, it is key to maintain the data sovereignty of their machines in order to be able to offer synergy effects for clients operating with a multitude of Schneider Electric machines.

In summary, Schneider Electric could expand these and other applications into a holistic cloud-based maintenance offering in which, for example, not only information on machine conditions is shared with clients but maintenance services can also be booked directly.

III. DATA UNDERSTANDING

The data understanding chapter is split into three sub-sections. In the first section, we analyzed the Schneider Electric raw dataset to get a better data understanding of the case and data features. Afterwards in section two, we assessed the data quality of the dataset. In the third section, we plotted several visualizations of the datasets to get a better glimpse at the differences and special characteristics of the two datasets.

A. Schneider Electric dataset

As a starting point for the project, Schneider Electric provided us with two separate datasets. Both datasets include data points containing the power consumption corresponding to various speeds and two types of motion. There are 45,000 records of power consumption for each of the 20 speed levels, measured at equal time intervals of 10ms. Therefore, the duration of each set of speed levels is 450 seconds, or 7.5 minutes. The speed levels are measured in rotations per minute (rpm). The slowest speed is 100 rpm, and the maximum speed is 2000 rpms, with a stable increase of 100 rpms, between the speed levels. With higher rpms the respective amplitude of power consumption in Watts gets higher. This structure holds for both datasets. The difference between the two datasets is that one consists of a continuous motion pattern and the other of a stepwise or discrete motion pattern. A speed of 0 rpm is possible in both datasets. Considering each set of rpms, the respective observations each form a time series.

The data points have been measured from a motion system based on a single carrier which represents a small part of a more complex system. The carrier moves horizontally within a limited range, going back and forth repeatedly. The carrier travels without load. Since the datasets consist of sensor data from a connected machine, we can assume it is highly available and real-time streaming data.

To sum this up, the provided data can be described as organized, useful, categorized and repetitive.

B. Data Quality

The dataset provided by Schneider Electric is complete and free of errors. Therefore, no cleaning is necessary, which is typically consuming a substantial part of the time in data analytics projects. The datasets could be read in exactly as they were provided from Schneider Electric. Generally, the upfront assumption we took is, that the provided data is relevant, up to date and sufficient to develop a predictive maintenance strategy. This assumption would be confirmed in later analysis as well.

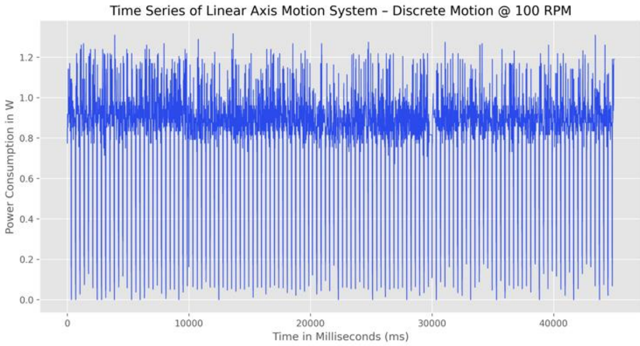


Figure 2: Time series of the linear axis motion system at 100 rpm (discrete motion) with the power consumption (own illustration)

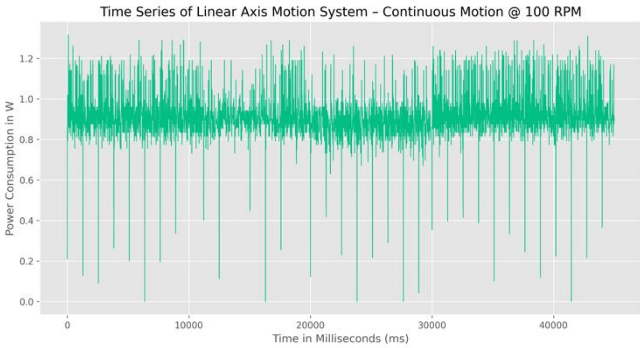


Figure 3: Time series of the linear axis motion system at 100 rpm (continuous motion) with the power consumption (own illustration)

Viewing each time series graphically, there are no indications of any trends in the data. The amplitude of each respective speed level also remains constant over time. This makes sense, because the motion pattern of the machine repeats itself and is not affected by external factors. For this reason, the provided data is most likely stationary. However, due to the direction change of the linear axis motion system, the respective power consumption decreases to zero at periodic intervals. As a result, the graphical representation indicates strong seasonality on a micro level. Seasonality refers to the predictability of data patterns that repeat themselves over time (Dudek, G., 2013).

In order to predict the future power consumption, we only consider the past power consumption as the single variable that is varying over time. No additional variables have been provided and thus no feature engineering techniques have been applied to enrich the datasets with new features. Each of the 20 rpm sets separately can therefore be described as a *univariate* time series.

C. *Problem formulation*

The given time series forecasting problem belongs to the domain of supervised learning. We have clear, labeled and structured numerical data that should be trained with a Deep Learning approach. The desired model should learn about the data and predict new, unseen inputs at best accuracy. The approach also includes the selection of the right Deep Learning models and tuning of the models` hyperparameters. Applying current knowledge to solve the time series problem, reevaluating and modifying said knowledge and adjusting the applied concepts is also an important aspect.

IV. Data Preparation

In the fourth chapter, we will present and define our approach in a technical sense. We outline how the data will be handled and what strategies we use. Second, we will describe how we prepared the data to fit it to a supervised Deep Learning model in order to enable optimal model performance.

A. *Defining the approach*

Deep Learning models can handle various types of data from unstructured and structured data to natural language, pictures, various data-formats to time series data (Manero Font, J., 2020). As mentioned above, our model will handle univariate past time series data to predict future power consumption data. When developing a forecasting model that uses historic data to obtain future predictions, it is possible to distinguish between two strategies: *single-step* (or one-step-ahead) forecasting and *multi-step* forecasting (Lim & Zohren, 2021). Single-step forecasting refers to an approach where the model learns from an input sequence and predicts the respective next single datapoint in the timeline. The single-step approach generally yields future predictions with higher accuracy than multi-step forecasts (Suradhaniwar, S., et al., 2021). Therefore, the informational usefulness of single-step forecasts is limited in most cases due to the mentioned short forecast horizon. A multi-step forecast approach addresses this by predicting a sequence of future datapoints as an output. This output sequence has a fixed length FH (the forecast horizon), which can be illustrated by the following formula:

$$[x_1, x_2, x_3, ..., x_n] \rightarrow [\hat{y}_{t+1}, \hat{y}_{t+2}, ..., \hat{y}_{t+FH}] \quad (1)$$

According to An et al. (2015), a multi-step-ahead prediction can also be achieved through combining multiple single-step forecasts through an *iterative* strategy. However, iterative methods will perhaps perform poorly on long forecast-horizons (Bontempi et al 2013). Developing a Deep Learning model that can predict a forecast horizon > 1 is generally less accurate than a single-step approach due to several factors, such as the accumulation of errors and the lack of precise information (Sorjamaa et al., 2007).

The length of the input-sequence used by the Deep Learning model is also highly relevant. The approach to this supervised Deep Learning problem involves a so-called *rolling-window* strategy. This rolling-window approach enables the Deep Learning model to stepwise 'slide over' the time series and predicting an output sequence as a result of an input sequence of a certain window-size. Each window is thereby a subset of the data with a predetermined and constant length and serves as the input for the model. At each iteration, this construct will move one step further and repeat its procedure. These steps will be replicated until the model has covered the entire training-dataset. Figure 4 shows a sketch of the idea behind this approach. In blue, the input sequence for the sliding-window approach is illustrated, and in orange, a hypothetical multi-step forecast-horizon is depicted.

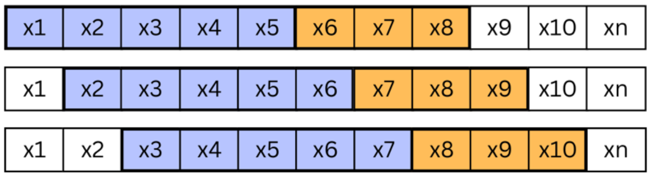


Figure 4: Exemplification of the idea behind the sliding-/moving-window approach (blue) and multi-step forecasting (orange) – 3 iterations are shown. (Own illustration)

Regarding Bengio et al. (2013), deep neural networks can learn about and gain a strong understanding on complex data. Therefore, while the sliding-window iterates through the training data, the model gains a comprehensive knowledge of the complex dependencies and characteristics of the input data. This special kind of multi-step approach, where an input sequence with a window-size > 1 directly predicts a whole sequence of predictions > 1, is also known as a *Multiple-Input Multiple-Output (MIMO)* strategy (Bontempi, G., et al., 2013).

The focus of the modeling and coding part of this paper will be the implementation of the mentioned MIMO strategy. In addition, we conducted a further-in-time prediction of additional 600 ms using a combination of the MIMO strategy and the recursive method. Instead of iterating over the results of single-step outputs to obtain a multi-step forecast, we iterate over outputs that already contain multiple predictions. This should result in a better performance than the standard recursive method because fewer iterations are required to produce the same number of future predictions. This should give a short outlook on how future data will look like. The general focus of our approach lies on the MIMO strategy, which uses a certain input window to predict a set of future predictions.

B. Preparing the data to supervised learning

In order to create and train a model, the dataset has to be split into three parts. The first and main part is the training set, which is provided to the model during the training process in order to learn the characteristics of the dataset. In addition to the training set, a separate validation set can also be introduced during the training process of the model. Based on the input from the training set, the model is trained to make predictions. Then it is given the input data of the validation set and compares its predictions to the provided output of the validation set. The test dataset is the one part of the whole dataset that should only be introduced to the model after the creation and training process. As the training process is the most important part, the training data makes up a major part of the whole dataset. Thus, typical training sizes are around 80 % and have been shown to result in better training performance and a more stable model than lower proportions (Nguyen Q., 2021). There are different procedures for how to split the data. However, for time series, the individual data points cannot be chosen randomly due to time-dependencies. Thus, we used the first 36,000 * 10ms as training, then the next 4,500 * 10ms as validation set. The remaining 4,500 * 10ms were kept as test data. The relative contributions are 80 %, 10 % and 10 % for training, validation and test set, respectively.

An additional part of the preparation for the modelling is data transformation. There are several techniques, like Normalization or Standardization. The purpose of these techniques is to simplify the data without changing its characteristics. This helps the model in the training process to learn faster and more precisely. Normalization rescales the range of the data to a chosen range, for example, between 0 and 1 (Garcia S., 2016). Standardization can be used for data following a Gaussian distribution. As it subtracts the mean and divides by the standard deviation, the transformed data has a mean of zero and a standard deviation of 1 and thus follows a standard normal distribution.

V. Deep Learning Modelling

A. Deep Learning Fundamentals

First, we explain how we have chosen the models and techniques. A theoretical description of the used models in the literature will be presented, followed by a more detailed characteristic of the ones we used. Lastly, we point out how we adjusted the models for best performance in order to be optimized for the application in a predictive maintenance strategy. The analyzed approaches include Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). The selection of the model is based on an assessment of the suitability of these approaches for Predictive Maintenance approaches according to a literature review (Abdelli et al., 2022; Mazzei & Ramjattan, 2022; Le et al., 2019; Naren & Subhashini, 2020; Olesen & Shaker, 2020; Rengasamy et al., 2020; Yu et al., 2020).

Starting with simpler models, there is a large number of techniques and models that can be used for the forecasting problem. Traditional stochastic models for time-series are Auto-regressive (AR) or Moving Average (MA). A well-known method is a combination of these two models, the "Auto-Regressive Integrated Moving Average (ARIMA)", which can be used for both univariate and multivariate datasets. With increasing computational power and the availability of open-source frameworks like TensorFlow (Abadi et al. 2015), more advanced techniques have evolved over time. This more complex part of machine learning is called Deep Learning as it employs the deep neural network (Kim, 2017). A neural network is a network of nodes, similar to neurons in the human brain. Neurons are the smallest functional unit of the nervous system. At each node, the weighted sum of the input signals is calculated. If the result exceeds a certain threshold, an output is generated (Seikel et al., 2020). An artificial neural network makes use of this structure, whereas a deep neural network is a further extension of the shallow neural network by containing two or more so-called hidden layers.

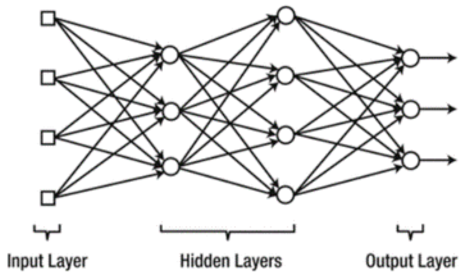


Figure 5: A layered structure of nodes (Kim, 2017)

Figure 5 shows the basic structure of a deep neural network, which consists of an input layer, several hidden layers and an output layer. The input layer receives input from outside the model, whereas the output layer transmits information to the outside. The layers in between are called hidden layers because they are not observable from the outside and only have connections within the network.

Going back into the micro-cosmic view of one single neuron helps to understand the learning process of a neural network. Figure 6 shows a single node receiving several inputs with the respective weights and, after calculation, passing through a transfer function.

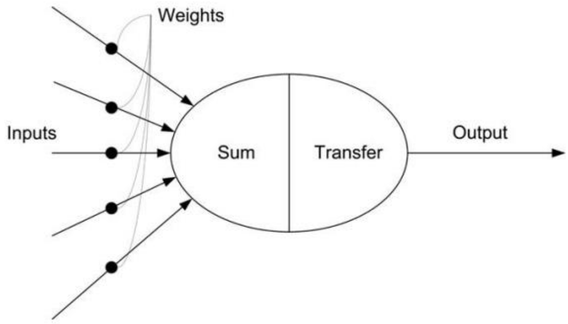


Figure 6: Structure of a single neuron: inputs are summed up with their respective weights and then passed through a transfer (Kavakioglu, 2019).

During the learning process, the connection weights are modified according to the mismatch between the generated and desired output which is also provided to the model in the case of supervised learning. In the case of unsupervised learning, the model receives no output beforehand. Another type of learning process is called reinforcement learning. In this case, instead of comparing the generated output to a given one, the model is being graded as being good or bad (Kavaklioglu, 2019).

B. *Deep Learning Layers*

Before explaining the Deep Learning architecture of the model used in the present study, this section briefly introduces the basic concepts of Deep Learning building blocks. Lim and Zohren (2021) distinguish between two neural networks (Lim & Zohren, 2021): CNN and RNN. CNNs are feed-forward models that are made of two separate networks, one that extracts features and the second as a classifier (Markiewicz et al., 2019). The feature extractor first creates feature maps of the original image within the convolutional layer and then reduces the image size within the pooling layer. For pooling, there exist several methods, e.g. mean pooling, where neighboring pixels are combined into a single pixel via a mean calculation. The final image is then fed into the classification network, which operates based on the features of this image. Traditionally made for image recognition, CNNs can also be used for forecasting problems, where “a convolution can be seen as applying and sliding a filter over the time series” (Ismail Fawaz et al., 2019).

In contrast to CNNs, RNNs are usually trained with Backpropagation, meaning that during learning, the model starts at the right-most cell of the network and moves backwards. Due to its capability of summarizing the past by having an internal memory state, RNNs have traditionally been developed for forecasting applications. The above-mentioned backpropagation however leads to one major limitations of classical RNNs: vanishing or exploding gradients (Serradilla et al., 2022, Marino D., 2016, Bengio 1994). To briefly summarize this problem, the gradient of the weights gets smaller with each propagation in the case of vanishing gradient. When the gradient decreases, learning slows down until it completely shuts down when the gradient becomes zero. The opposite would be exploding gradient where the gradient becomes larger for every backpropagation. To solve this issue, LSTM, which is essentially a model that has been shown to perform well on problems with long time lags, was developed. (Serradilla et al., 2022; Hochreiter & Schmidhuber, 1997; Hochreiter 1998).

Regarding the architecture of LSTM units, gates are used to modulate the data transfer to the successive layer. Each LSTM has three kinds of gates: a Forget, a Memory and an Output Gate. The Forget Gate decides how much of the input data should be kept and how much should be disposed, whereas the “Memory Gate chooses which new data need to be stored in the cell” (Siami-Namini, 2018). Finally, the Output Gate is responsible for what will be passed on to the following unit.

The GRU evolved from the classical RNN in 2014 much like the LSTM but with a slightly simpler architecture within each cell (Cho et al 2014). Whereas the LSTM unit also has separate memory cells, the GRU only “has gating units that modulate the flow of information inside the unit” (Chung, 2014), namely an Update and a Reset Gate. Basically, the Update Gate decides which part of the input needs to be kept, whereas the Reset Gate determines which part to forget.

Regardless of whether the decision is made for CNN or RNN or even a combination of both, those are the core elements of a Deep Learning model. However, other building blocks of such a model have to be considered as well. First of all, the model is initialized by a so-called Sequential. This is not a layer itself, but a method to stack several layers one at a time in a linear fashion.

In addition, between convolutional or recurrent layers, dense layers can be implemented. Dense layers, also called fully connected layers, are connected to every input neuron from the previous layer, hence the name dense. Thus, fully connected layers can “extract the global features” (Xu et al., 2018) and “are essential in prediction tasks and can easily change the length of the output sequence” (Wan et al., 2019).

However, extracting too many features, especially in noisy data, can lead to overfitting. This can be delayed by preventing neurons from overspecializing. That is where the dropout layer can be used, as it temporarily disconnects some of the neurons from the previous layer (Glassner, 2021).

Furthermore, the so-called repeat vector has to be named, as it is also used in the present study. A repeat vector simply repeats the sequence of the previous layer and can be used between two LSTM-layers as part of an encoder-decoder model, as used by Essien et al. (2021) in a Deep Learning model to predict urban traffic flow. An encoder-decoder model was also used by Mehtab et al. (2021) to predict stock prices. The authors built a model with two LSTM-layers with a repeat vector in between. The first LSTM layer reads and encodes the input sequence, whereas the second layer then reads the encoded input and makes a prediction for each element in the output. This structure was followed by a *TimeDistributed* layer ”that packs the interpretation layer and the output layer in a time-synchronized manner” (Mehtab et al. 2021). This helps the decoder understand the context of the output sequence by reusing the same weights.

C. *Model Architecture*

A similar architecture has been implemented for the MIMO forecast in the present study (see Fig. 4). The first LSTM (GRU in the second model) layer consists of 128 nodes and acts as a decoder that reads the input sequence having the shape (40, 20). This shape is due to the fact that we chose an input window size of 40 data points for each of the 20 different rpms, respectively. The output shape of the repeat vector layer (20, 128) corresponds to the 20 timesteps as the forecast horizon in the output sequence and the 128 features being extracted. The output of the second LSTM (or GRU) is then passed through a Dense layer using a time distribution in order to interpret the output in a time-synchronized manner, as already explained. The following and last Dense layer produces the final output with the right shape for 20 forecasts and 20 features (rpms). For the encoder, we used the rectified linear unit (ReLU) activation function. ReLU is basically a calculation that returns the input directly or zero if the input is less than or equal to zero. The Mean Absolut Error (MAE) was used as the loss function and Adam as optimizer. Adam is “an algorithm for first-order gradient-based optimization” (Kingma 2014).

Figure 7 summarizes the general modeling workflow. It consisted of the data preparation and preprocessing, the model selection, including the configuration of the hyperparameters, the inverse-processing, and lastly, the model performance evaluation. Regarding the model input, we decided to feed all 20 speed categories into the model. First, the dataset was split into training, validation and test datasets with a share of 80 %, 10 % and 10 %, respectively.

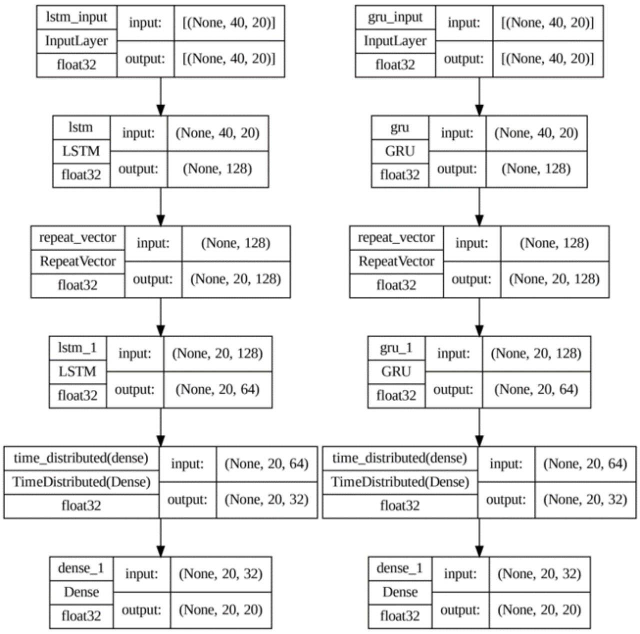


Figure 7: LSTM and GRU Layers (tf.keras.utils.plot_model)

To avoid distortion caused by the different amplitudes of the speed categories, we normalized all data-streams using the scikit-learn function MinMaxScaler (Pedregosa F. et al., 2012) to obtain the best model performance. After normalization, the sliding windows approach was applied to the data, creating a 3D vector with [samples, timestamps, features]. 40 and 20 timestamps were chosen as the window size of the input (lookback) and output (forecast-horizon), respectively. We implemented all models using Python 3.9.12 on the TensorFlow 2.9.2 and Keras 2.9.0 frameworks. The models were trained in the Google Colab cloud environment using an Intel Xeon CPU running at 2.2 GHz and 12.7 GB of RAM.

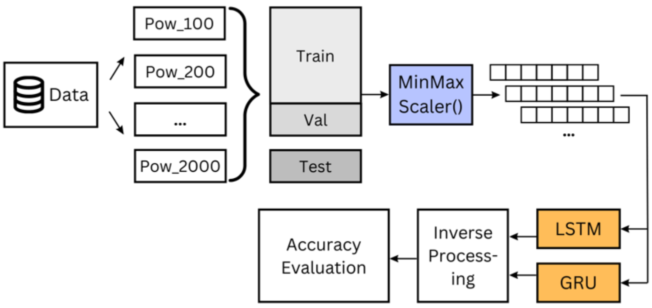


Figure 8: Modeling Pipeline

VI. Results

A. Evaluation

As already mentioned, the data we obtained from Schneider Electric contained separate datasets for both, discrete and continuous motion patterns of the machine. For the discrete and continuous motion datasets, LSTM and GRU models have been trained. This approach resulted in four separate models: one LSTM and one GRU model for each type of motion. Table 1 shows the evaluation metrics, namely the MAE and Mean Squared Error (MSE) for these models. It has to be mentioned that these error KPIs have been derived prior to the postprocessing, thus are still normalized, and are calculated over all forecast horizons (from t+1 to t+20). This table should just serve as a summary of the models' performances. Both models perform very similar for both, discrete and continuous motion datasets.

Model	MAE	MSE
LSTM Discrete	0.087598	0.024937
GRU Discrete	0.089566	0.023685
LSTM Cont.	0.086361	0.027852
GRU Cont.	0.080795	0.025385

Table 1: Error metrics of LSTM and GRU applied to both, discrete and continuous (Cont.) motion data

To provide more insights, Table 1 shows the development of the loss metrics over the training process of 20 epochs. It has been shown that 20 epochs are sufficient for training the model. A higher number did not lead to significantly better loss metrics, especially the validation loss. Beginning with epoch 5, the two lines already diverge, with the validation loss having a less steep curve. This indicates that the model is still learning from the training set, although its performance does not get better. In this case, setting the epochs to a very large number would lead to overfitting, where the model learns the training set too well and captures the noise in the data. To prevent overfitting even for 20 epochs, a stop function has been implemented when the validation loss does not improve after a patience period of 5 epochs.

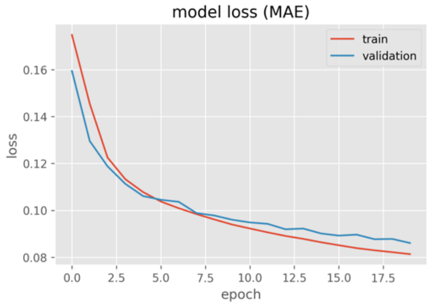


Figure 9: Development of the training and validation loss (MAE) over the 20 epochs. Example from GRU model.

Figure 6 allows a more in-depth comparison of the prediction accuracy of the LSTM for discrete motion data. The individual graphs display the development of the Root Mean Square Error (RMSE) when increasing the forecast horizon within the individual velocities. As one might imagine, the RMSE increases the further into the future the model tries to predict. Interestingly though, the patterns are different. Especially for the higher velocities, we discovered a sharp increase within the first few forecast windows and after that, a relatively stable error. This shows that carefully selecting the forecast windows can further improve accuracy.

B. Validation

Several tests have been conducted in order to validate that the model trains each of the 20 time series separately without any kind of cross-training or interference.

(1) For the first test, the model was trained on a dataset containing 19 random time series and one real time series. The random time series were generated between a scale of 1 to 30, so significantly greater than the actual data. All other parameters of the code remained unchanged. After testing, the results showed that the

unchanged time series was predicted as accurately as without the presence of the 19 random time series. Therefore, we can assume, that the one real time series did not get influenced by the other random time series during the training phase.

(2) For the second test, the model was trained on the training and validation datasets as usual and then tested on the dataset including the 19 random time series and the one real time series, as in the first test. The results showed that the unchanged time series was predicted with no loss in accuracy again.

(3) The third test involved investigating the positional changes of the time series within the input-matrix that feeds the model. For example, the effect of swapping two columns once training has been completed. The results however showed that the accuracy of the predictions depends on the unchanged positions of the input data. Therefore, we can conclude that for highly accurate predictions, the position of the input time series must match the position in the data it was trained on.

In summary, we can say that the model relies on an input of 20 time series but can predict a single time series with high accuracy as well. Simply by inserting random numbers in the remaining space inside the required 20-column input matrix, the model could make individual predictions for certain rpms of the machine, that were not filled with random numbers. Additionally, the input data should be in the same position that it will be tested on later. In case the model should be capable of classifying itself, which rpm certain input data have, a method could be implemented, which assigns the rpm according to its unique amplitude to the right columns in the input matrix.

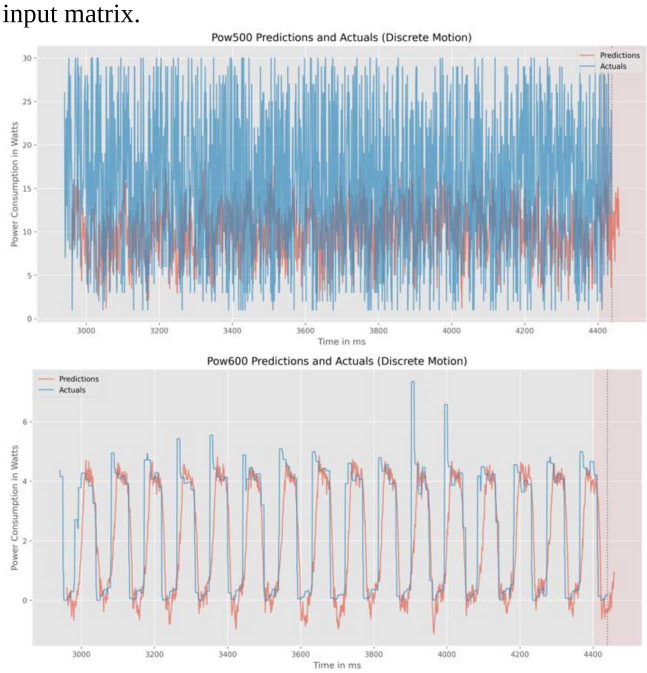


Figure 10: Model predictions based on 19 random and one actual column training and testing. The unmodified column, Pow600, still shows accurate results (Test 1)

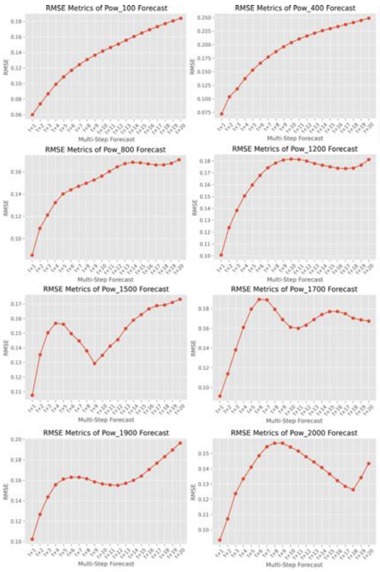


Figure 11: RMSE over the Forecast Horizon for each velocity. Model: LSTM. Dataset: Discrete motion

VII. DEPLOYMENT

The last step of the CRISP-DM process is the deployment phase. The objective in this section is that the gained knowledge and developed processes will be transferred to the customer in an accessible manner. To achieve this, a detailed project report is provided. We aimed to ensure that all business and technical assumptions, approaches, and limitations were well explained and that the results of our deep learning model are reproducible. To make the developed code as easily accessible for clients as possible, a readme file with comprehensive comments is included as well.

A. Cloud embedment

In order to make sure that the developed power consumption forecasting model for predictive maintenance using a Deep Learning algorithm is working while exposing it to a real use case, we suggest the implementation of the model in a fully managed cloud dataflow service. Possible examples are Google Cloud Dataflow within the Google Cloud Platform, Azure Data Factory from Microsoft or AWS Data Pipeline from Amazon Web Services. Paolanti et al. (2018) provide an exemplary general scheme of an overall cloud architecture for a prediction model based on Azure (Paolanti et al., 2018). A detailed overview of an exemplary embedding of our power consumption forecasting model in a cloud environment using the example of Google Cloud can be found in Appendix B (Google Cloud, 2023). Our model would be integrated into the VertexAI module, while at the same time, the streaming data of the machines would be processed with Google Cloud Dataflow. The use of Dataflow and VertexAI for deploying the deep learning model provides several benefits for Schneider Electric. This cloud connection safeguards the necessary computing power and is also flexibly expandable and scalable. The potential and limitations of cloud computing are assessed in detail by Teoh et al. (2020). Since this service is built on Google Cloud, which provides a highly scalable infrastructure which ensures that the model is able to handle an increase in the amount of real time data without any computational issues. Another key benefit of using Dataflow and VertexAI is the reliability of the cloud service as well as the access to the latest Machine

Learning technologies and the possibility to deploy custom models. Moreover, Dataflow and VertexAI connected with integrated visualization tools provide a user-friendly interface, which is key for the acceptance in the maintenance team. Finally, Dataflow and VertexAI provide a cost-effective solution for deploying the deep learning model, as they allow the company to pay only for the resources they use.

Despite a reliable Cloud embedment of the model, a stable connection of both the machines and sensors as well as the smart devices of the maintenance team is necessary. Therefore, various Application Programming Interfaces (APIs) have to be clearly defined. One is the interface between the machine and the chosen cloud service for the upload of the gathered machine data. The other interface is characterized by the way the operator receives back the automated recommendations for suggested maintenance activities. Here, a good and user-friendly front-end on the respective smart device of the operator will be important so that the system will be accepted and used by the maintenance team. Additionally, it is key, that such a model implementation into the daily is also supported by the whole company. There is a need for training not only on the maintenance team side but also on the production team side. Furthermore, the IT department must support the implementation not only in the rollout phase of a pilot project use case but also during the whole deployment. The corporate IT department needs to be the point of contact for technical support during the whole product lifecycle.

B. *Implementation planning and performce monitoring*

Another important factor for a successful deployment is a thorough planning and monitoring of the project together with a sustained emphasis on the importance and a continuous support from the management of the company. Here, a thorough planned implementation plan is the key to a successful rollout. This implementation plan should consist out of specifications regarding the following aspects: Hardware and Software requirements, meaning necessary servers and storage, data requirements, needed resources as for example personnel and expertise, together with the respective budget, training and testing with performance evaluations, a plan to minimize the impact of potential risks, a plan for the maintenance and retraining of the model over time and active communication together with stakeholder management.

In addition, it is important to measure the success of a cloud-based Predictive Maintenance service in order to be accountable. For that purpose, Aksa et al. (2021) suggest a list of various key performance indicators such as overall equipment effectiveness, quality, cycle time ratio, availability, maintenance cost and equipment life (Aksa et al., 2021).

VIII. Business Implications

From our final power consumption forecasting model with a Deep Learning approach, we were able to derive business implications in the form of key insights and limitations.

A. *Key insights*

(1) We derived a Deep Learning model that is not only able to predict the power consumption of a motion system based on a linear axis with a single carrier but is also generalizable to other power consumption patterns. The model detects abbreviations from a predicted value in comparison to an actual value based on historical training.

(2) This mechanism can be used as an early warning system, for example in predictive maintenance in order to optimize the machine reliability and availability. This results in increased productivity and saved costs as the need for expansive repairs will be reduced while efficiency will be increased through less downtime (Selcuk, 2015). Reducing downtime, improving efficiency and increasing machine availability results in an increased revenue.

(3) With our forecasting model, the complete maintenance procedure can be optimized. The model allows for improvements from reactive maintenance towards a risk-based Prescriptive Maintenance approach. This improves the security level and saves unnecessary and redundant maintenance on the one hand and the costs of unexpected machine downtime on the other. Schneider Electric can sell their machines as smart or IoT machines with integrated predictive maintenance. This can be translated in additional revenue streams through leasing or subscription models of this service.

(4) With our cloud-based prediction model, a real-time condition monitoring of industrial machines becomes broadly available for the industry. Because of the cloud architecture, scalability is not an issue and IT-infrastructure costs can be better estimated.

(5) Further potential to be exploited lies in the other data that industrial machines are generating. This data could be used and analysed with Big Data methods to generate even more insights into the use and wear of a machine over its complete lifetime. Also, the development of software for future machines can be improved due to the established Big Data repository.

(6) Being able to maintain the Data sovereignty will be key for the offering of maintenance services.

(7) Improved sustainability: Our forecasting model will help Schneider Electric to improve sustainability by reducing energy consumption, decreasing carbon emissions, and extending the lifespan of machines (Henning et al., 2021).

B. *Limitations*

Although our model provides Schneider Electric with results ready for deployment, there are limitations that should be considered for further investigation.

(1) Data-centric improvements: The prediction output of our model depends on the quality of the provided data. It can be possible that sensors are inaccurate in measuring the velocity of the carrier (Pech et al., 2021). Furthermore, there may be other factors that slow or accelerate the carrier. In addition, features can and should be extracted and added in order to improve the model's performance. Examples of useful features could be the range of the dataset (without providing future features to present data) or a moving average. The present models did not exploit the opportunity of feature engineering, although the applicability of the model would benefit from more measured features, such as for example the temperature of the machine, noise, vibration any many more (Aksa et al.,

2021; Traini et al., 2020; Gouarir et al., 2018; Paolanti et al., 2018; Selcuk, 2015).

(2) Model-centric improvements include methods like hyperparameter tuning. During the testing phase, RandomSearch has been applied to some metrics of the model, including the number of units per layer, dropout and learning rates. However, this only led to minor improvements despite being very time-consuming. More important parameters that can certainly be optimized, are the layers of the model and the number of epochs, which have been fixed prior to tuning and thus have not been optimized.

(3) The model was only trained on the given velocities, from 100 to 2000 rpm (step: 100 rpm). Thus, predictions can only be made based on those given velocities. In case the machines should be able to run at intermediate, higher, or lower velocities than those trained ones, no valid outcomes can be predicted. However, for new arising velocities, adjustments or updates can easily be made.

(4) The data points have been collected every 10 ms, so with a frequency of 100 Hz. During the data understanding phase, we observed a lot of repeating rows. This implies that the frequency of the velocity sensor is higher than the change in velocity itself, resulting in a high proportion of redundant data. Thus, sampling to a certain degree could get rid of redundancy without losing relevant information. This could reduce the time consumption of the prediction process and consequently, increase the usefulness of the forecast by getting the results faster.

(5) As already mentioned, only the machine’s velocity was fed into the model. Besides extracting and adding features of the velocity dataset, including further information would be helpful, especially in the case of predictive maintenance. Including information about the current operating status or date of the last maintenance, to name a few, would certainly be helpful to get deeper insights about the machine’s condition.

(6) The model is currently not able to measure how critical an anomaly or the detection of a deviation is. A threshold for the criticality is open to be defined. This accuracy is of utmost importance for the automated recommendation that is sent out to the on-site operator. For the model to be accepted and really being used in practice, a high percentage of relevant detections in comparison to false alarms is key.

(7) Another feature that is currently not implemented but could add significant value to the model is the ability to learn which recommendations based on which detections were relevant to the operator and which were not. This could be done by developing a Deep Reinforcement Learning Model that evaluates and learns from a response of the operator regarding the helpfulness of the recommendation.

(8) The model’s predictions rely on the correct positional information of the input data and cannot be interchanged later. This can be fixed by implementing a method that correctly assigns each input stream to a specific rpm and places it in the correct column of the input matrix for later predictions. Because every rpm has a rather unique amplitude, such a classifier should work reliable.

IX. CONCLUSION AND OUTLOOK

The power consumption model using a Deep Learning approach enables Schneider Electric to offer a holistic product portfolio. They are not only selling a product like a machine to a customer, but they are also offering a complete service on a subscription basis. Clients can rent machines and Schneider Electric is carrying out the complete maintenance business for the client. But Schneider Electric sells not only a smart maintenance service, but also secures additional revenue as they are clearly capable of measuring the improvements in the overall equipment effectiveness of their machines. This can be translated into higher productivity and therefore reduced costs for the client. Schneider Electric might be able to agree on a flexible payment model that allows them to receive a share of these savings. This results in a huge business potential for Schneider Electric, which might lead to a competitive advantage in comparison to other manufacturers of industrial automation and control products.

The models can predict the power consumption for up to 200 ms with a velocity-dependent accuracy. Especially at higher velocities, the accuracy increases. Regarding its use in the field of predictive maintenance, the exact match between actual and predicted values is of less importance compared to the correct range and shape of the line, which our models have shown to be capable of. With the application of the sliding window approach, we were able to compare the accuracy of different forecast horizons. Unsurprisingly, the further the forecast goes into the future, the higher are the error rates. Interestingly though, the pattern of this deterioration along the forecast horizon was not uniform for all velocities.

In summary, for better applicability and accuracy, more data regarding the various characteristics of the machine, such as the current operating status, would be beneficial. In addition, the model could be extended with modules measuring the level of deviation and therefore, the relevance of a detected anomaly.

In a next step, the model could be enriched with a module that trains itself whether a detection is relevant based on the operator's response. To achieve this, a deep reinforcement learning model could be trained regarding the relevance and correctness of the recommendation of the anomaly detection. Every time our power consumption forecasting model detects a deviation from the expected amount, the operator is asked whether this detection was a relevant case which required an action from the operator or if the machine is performing normally. With this additional deep reinforcement learning model, we empower our forecasting model to only report important deviations and therefore reduce the number of false alarms. This leads to an increased acceptance and creditability of the forecasting model as the operators see that the recommendations are accurate and reliable. Over time, a resilient knowledge base including detailed monitoring of a specific machine's operating condition and its wear parts, can be developed. This knowledge can also be sold to other companies that want to improve their maintenance business.

The gained knowledge could also be used for electricity peak shaving, which allows companies to minimize energy costs (Uakhitova, 2022; La Nieta, 2021; Nisa & Kuan, 2021; Lv et al., 2020).

ACKNOWLEDGMENT

We thank Schneider Electric for providing us with real and relevant data. We appreciate your help and discussions in the Q&A sessions during the seminar. These were helpful to clarify open questions. Moreover, we thank Sebastian Junker and Niklas Stepanek for the continuous support and help throughout the course of this inspiring hands-on seminar.

REFERENCES

[1] Abdelli, K., Grieser, H., & Pachnicke, S. (2022). A Machine Learning-Based Framework for Predictive Maintenance of Semiconductor Laser for Optical Communication. Journal of Lightwave Technology, 40(14), 4698–4708. https://doi.org/10.1109/JLT.2022.3163579.

[2] Akiyoshi, H., Hiraki, E., Tanaka, T., Okamoto, M., Matsuo, T., & Ochi, K. (2014). Peak Power Shaving of an Electric Injection Molding Machine With Supercapacitors. Peak Power Shaving of an Electric Injection Molding Machine with Supercapacitors, 50(2), 1114–1120. https://doi.org/10.1109/TIA.2013.2272433.

[3] Barriball, E., Cox, M., Snakur, A., & Wijpkema, J. (2017). Smarter choices: Raise asset availability and reduce operating costs: New analytics techniques can help companies keep more of their assets working more of the time, while reducing inventory levels and lifecycle costs. McKinsey & Company. https://www.mckinsey.com/capabilities/operations/our-insights/smarter-choices-raise-asset-availability-and-reduce-operating-costs (Operations Extreanet).

[4] Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 35(8), 1798-1828.

[5] Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks, 5(2), 157-166. https://doi.org/10.1109/72.279181

[6] Berg, A. (2020). Industrie 4.0 – so digital sind Deutschlands Fabriken. Bitkom Research 2020. https://www.bitkom.org/sites/default/files/2020-05/200519_bitkomprasentation_industrie40_2020_final.pdf

[7] Bitkom. (19. Mai, 2020). Was sind aus Sicht Ihres Unternehmens die wichtigsten Vorteile von Künstlicher Intelligenz im Kontext von Industrie 4.0? [Graph]. In Statista. Zugriff am 29. Januar 2023, von https://de.statista.com/statistik/daten/studie/990505/umfrage/umfrage-zu-vorteilen-kuenstlicher-intelligenz-in-deutschen-industrieunternehmen/

[8] Bontempi, G. (2008). Long term time series prediction with multi-input multi-output local learning. Proc. 2nd ESTSP, 145-154.

[9] Bontempi, G., Ben Taieb, S., & Le Borgne, Y. A. (2013). Machine learning strategies for time series forecasting. Business Intelligence: Second European Summer School, eBISS 2012, Brussels, Belgium, July 15-21, 2012, Tutorial Lectures 2, 62-77.

[10] Bradbury, S., Carpizo, B., Gentzel, M [Matt], Horah, D., & Thibert, J [Joël] (2018, October 4). Digitally enabled reliability: Beyond predictive maintenance: To capture everything digital can offer in increasing reliability and reducing costs, companies should boost their digital-maintenance ambitions. McKinsey & Company. https://www.mckinsey.com/capabilities/operations/our-insights/digitally-enabled-reliability-beyond-predictive-maintenance

[11] Chaerun Nisa, E., & Kuan, Y.-D. (2021). Comparative Assessment to Predict and Forecast Water-Cooled Chiller Power Consumption Using Machine Learning and Deep Learning Algorithms. Sustainability, 13(2), 744. https://doi.org/10.3390/su13020744

[12] Cho, K., Merrienboer, B., Bahdanau, D., & Bengio, Y. (2014). On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. https://doi.org/10.3115/v1/W14-4012.

[13] Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.

[14] Cortes, T., Decaix, G., Hansmann, T., & Tan, Khoon Tee, Zhou, Yi (2021, July 22). Prediction at scale: How industry can get more value out of maintenance: Machines can now tell you when they aren't feeling well. The challenge for today's industrial players lies in applying advanced predictive-maintenance technologies across the full scope of their operations. Operations Practice. McKinsey & Company. https://www.mckinsey.com/capabilities/operations/our-insights/prediction-at-scale-how-industry-can-get-more-value-out-of-maintenance.

[15] Dabbagh, M., Hamdaoui, B., & Rayes, A. (2020). Peak Power Shaving for Reduced Electricity Costs in Cloud Data Centers: Opportunities and Challenges. Peak Power Shaving for Reduced Electricity Costs in Cloud Data Centers: Opportunities and Challenges, 34(3), 148–153. https://doi.org/10.1109/MNET.001.1900329

[16] Dalzochio, J., Kunst, R., Pignaton, E., Binotto, A., Sanyal, S., Favilla, J., & Barbosa, J. (2020). Machine learning and reasoning for predictive maintenance in Industry 4.0: Current status and challenges. Computers in Industry, 123, 103298. https://doi.org/10.1016/j.compind.2020.103298.

[17] Decaix, G., Gentzel, M [Matthew], Luse, A., Neise, P., & Thibert, J [Joel] (2021, April 23). A smarter way to digitize maintenance and reliability: In heavy-equipment maintenance, digital and analytics tools are struggling to live up to their promise. Here's how to help them pull their weight. Operations Practice. McKinsey & Company. https://www.mckinsey.com/capabilities/operations/our-insights/a-smarter-way-to-digitize-maintenance-and-reliability.

[18] Dennis, M., Velayudam, C., Subramanian, A., & Choudhary, S. (2017). Using Predictive Maintenance of Industrial Assets: Your Starting Point to the Digital Manufacturing Journe. Capgemini. https://www.capgemini.com/wp-content/uploads/2017/05/using-predictive-maintenance-of-industrial-assets-your-starting-point-to-the-digital-manufacturing-journey.pdf.

[19] Dudek, G. (2013). Forecasting time series with multiple seasonal cycles using neural networks with local learning. In Artificial Intelligence and Soft Computing: 12th International Conference, ICAISC 2013, Zakopane, Poland, June 9-13, 2013, Proceedings, Part I 12 (pp. 52-63). Springer Berlin Heidelberg.

[20] Elahi, M., Afolaranmi, S. O., Mohammed, W. M., & Martinez Lastra, J. L. (2022). Energy-Based Prognostics for Gradual Loss of Conveyor Belt Tension in Discrete Manufacturing Systems. Energies, 15(13), 4705. https://doi.org/10.3390/en15134705.

[21] Essien, A., Petrounias, I., Sampaio, P., & Sampaio, S. (2021). A deep-learning model for urban traffic flow prediction with traffic events mined from twitter. World Wide Web, 24(4), 1345-1368. https://doi.org/10.1007/s11280-020-00800-3.

[22] Fausing Olesen, J., & Shaker, H. R. (2020). Predictive Maintenance for Pump Systems and Thermal Power Plants: State-of-the-Art Review, Trends and Challenges. Sensors (Basel, Switzerland), 20(8). https://doi.org/10.3390/s20082425.

[23] Fraunhofer IPA (2019). Smart Predictive Maintenance. https://www.ipa.fraunhofer.de/content/dam/ipa/de/documents/Kompetenzen/Nachhaltige-Produktion-und-Qualitaet/PB_100_144_Smart_Predictive_Maintenance_web_offen.pdf.

[24] García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., & Herrera, F. (2016). Big data preprocessing: methods and prospects. Big Data Analytics, 1(1), 9. https://doi.org/10.1186/s41044-016-0014-0

[25] Gianoglio, C., Ragusa, E., Gastaldo, P., Gallesi, F., & Guastavino, F. (2021). Online Predictive Maintenance Monitoring Adopting Convolutional Neural Networks. Energies, 14(15), 4711. https://doi.org/10.3390/en14154711

[26] Glassner, A. (2021). Deep Learning: A Visual Approach. No Starch Press. https://books.google.de/books?id=NgTyDwAAQBAJ

[27] Google Cloud. (2023, January 13). Dataflow | Google Cloud. https://cloud.google.com/dataflow?hl=de.

[28] Henning, S., Hasselbring, W., Burmester, H. et al. Goals and measures for analyzing power consumption data in manufacturing enterprises. J. of Data, Inf. and Manag. 3, 65–82 (2021). https://doi.org/10.1007/s42488-021-00043-5.

[29] Hochreiter, S. (1998). The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 06(02), 107-116. https://doi.org/10.1142/s0218488598000094

[30] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.

[31] Holst, & Lennard. (2022). Instandhaltung als Enabler von Industrie 4.0: Predictive Maintenance & Smart Maintenance. https://www.fir.rwth-aachen.de/beratung/service-instandhaltung/predictive-maintenance-smart-maintenance/

[32] Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L., & Muller, P.-A. (2019). Deep learning for time series classification: a review.

Data Mining and Knowledge Discovery, 33(4), 917-963. https://doi.org/10.1007/s10618-019-00619-1.

[33] Jian Qin, Y. L. (2017). 2017 13th IEEE International Conference on Automation Science and Engineering (CASE): 20-23 Aug. 2017. IEEE. http://ieeexplore.ieee.org/servlet/opac?punumber=8244402.

[34] Kavaklioglu, K., Ceylan, H., Ozturk, H. K., & Canyurt, O. E. (2009). Modeling and prediction of Turkey's electricity consumption using Artificial Neural Networks. Energy Conversion and Management, 50(11), 2719-2727. https://doi.org/https://doi.org/10.1016/j.enconman.2009.06.016.

[35] Kiangala, K. S., & Wang, Z. (2020). An Effective Predictive Maintenance Framework for Conveyor Motors Using Dual Time-Series Imaging and Convolutional Neural Network in an Industry 4.0 Environment. IEEE Access, 8, 121033–121049. https://doi.org/10.1109/ACCESS.2020.3006788.

[36] Kim, P. (2017). Deep Learning. In P. Kim (Ed.), MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence (pp. 103-120). Apress. https://doi.org/10.1007/978-1-4842-2845-6_5.

[37] Kingma, D., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.

[38] La Nieta, A. A. S. d., Ilieva, I., Gibescu, M., Bremdal, B., Simonsen, S., & Gramme, E. (2021). Optimal midterm peak shaving cost in an electricity management system using behind customers' smart meter configuration. Applied Energy, 283, 116282. https://doi.org/10.1016/j.apenergy.2020.116282.

[39] Le, T., Vo, M. T., Vo, B., Hwang, E., Rho, S., & Baik, S. W. (2019). Improving Electric Energy Consumption Prediction Using CNN and Bi-LSTM. Applied Sciences, 9(20), 4237. https://doi.org/10.3390/app9204237.

[40] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

[41] Lim, B., & Zohren, S. (2021). Time-series forecasting with Deep Learning: a survey. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 379(2194), 20200209. https://doi.org/doi:10.1098/rsta.2020.0209

[42] Lorenz, M., Küpper, D., Rüßmann, M., Heidemann, A., & Bause, A. Time to Accelerate in the Race Toward Industry 4.0.

[43] Lv, J., Peng, T., Zhang, Y., & Wang, Y. (2021). A novel method to forecast energy consumption of selective laser melting processes. International Journal of Production Research, 59(8), 2375–2391. https://doi.org/10.1080/00207543.2020.1733126.

[44] Manero Font, J. (2020). Deep learning architectures applied to wind time series multi-step forecasting.

[45] Manyika, J., Chui, M., Bisson, O., Woetzel, J., Dibbs Richard, Bughin Jacques, & Aharon, D. (2015). The Internet of Things: Mapping the value beyond the hype: Executive Summary. McKinsey & Company. https://www.mckinsey.com/~/media/mckinsey/industries/technology%20media%20and%20telecommunications/high%20tech/our%20insights/the%20internet%20of%20things%20the%20value%20of%20digitizing%20the%20physical%20world/unlocking_the_potential_of_the_internet_of_things_executive_summary.pdf (McKinsey Global Insititue).

[46] Markiewicz, M., Wielgosz, M., Bochenski, M., Tabaczynski, W., Konieczny, T., & Kowalczyk, L. (2019). Predictive Maintenance of Induction Motors Using Ultra-Low Power Wireless Sensors and Compressed Recurrent Neural Networks. IEEE Access, 7, 178891–178902. https://doi.org/10.1109/ACCESS.2019.2953019.

[47] Mazzei, D., & Ramjattan, R. (2022). Machine Learning for Industry 4.0: A Systematic Review Using Deep Learning-Based Topic Modelling. Sensors, 22(22), 8641. https://doi.org/10.3390/s22228641

[48] Mehtab, S., Sen, J., & Dutta, A. (2021, 2021//). Stock Price Prediction Using Machine Learning and LSTM-Based Deep Learning Models. Machine Learning and Metaheuristics Algorithms, and Applications, Singapore.

[49] Naren, R., & Subhashini, J. (2020). Comparison of Deep Learning models for predictive maintenance. IOP Conference Series: Materials Science and Engineering, 912(2), 22029. https://doi.org/10.1088/1757-899X/912/2/022029

[50] Nguyen, K. T., & Medjaher, K. (2019). A new dynamic predictive maintenance framework using Deep Learning for failure prognostics. Reliability Engineering & System Safety, 188, 251–262. https://doi.org/10.1016/j.ress.2019.03.018.

[51] Nguyen, Q. H., Ly, H.-B., Ho, L. S., Al-Ansari, N., Le, H. V., Tran, V. Q., Prakash, I., & Pham, B. T. (2021). Influence of Data Splitting on Performance of Machine Learning Models in Prediction of Shear Strength of Soil. Mathematical Problems in Engineering, 2021, 4832864. https://doi.org/10.1155/2021/4832864.

[52] Osterwalder, Alexander & Pigneur, Yves. (2010). Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers.

[53] Paolanti, M., Romeo, L., Felicetti, A., Mancini, A., Frontoni, E., & Loncarski, J. (2018). Machine Learning approach for Predictive Maintenance in Industry 4.0. In 2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA): 2-4 July 2018 (pp. 1–6). IEEE. https://doi.org/10.1109/MESA.2018.8449150

[54] Paquin, R. (2014). Asset Management: The Changing Landscape of Predictive Maintenance. Aberdeen Group. https://web1.e-techservices.com/wp-content/uploads/2017/02/YTL03210USEN.pdf

[55] Pech, M., Vrchota, J., & Bednář, J. (2021). Predictive Maintenance and Intelligent Sensors in Smart Factory: Review. Sensors, 21(4), 1470. https://doi.org/10.3390/s21041470

[56] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., & Louppe, G. (2012). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12.

[57] Ran, Y., Zhou, X., Lin, P., Wen, Y., & Deng, R. (2019, December 12). A Survey of Predictive Maintenance: Systems, Purposes and Approaches. http://arxiv.org/pdf/1912.07383v1

[58] Rengasamy, D., Jafari, M., Rothwell, B., Chen, X., & Figueredo, G. P. (2020). Deep Learning with Dynamically Weighted Loss Function for Sensor-Based Prognostics and Health Management. Sensors (Basel, Switzerland), 20(3). https://doi.org/10.3390/s20030723

[59] Hyndman, R. J., & Athanasopoulos, G. (2018). Forecasting: Principles and Practice. https://research.monash.edu/en/publications/forecasting-principles-and-practice-2

[60] Schreiner, J., & Mundt, E. (2020). Was ist Predictive Maintenance? Definition, Anwendung und Beispiele: IoT-Basics. Industry of Things, 31.01.2020. https://www.industry-of-things.de/was-ist-predictive-maintenance-definition-anwendung-und-beispiele-a-693842/

[61] Schneider ELectric. (2021). Predictive Maintenance Solution with EcoStruxure Plant and Senseye. https://exchange.se.com/

[62] Seikel, J. A., Konstantopoulos, K., & Drumright, D. G. (2020). Neuroanatomy and Neurophysiology for Speech and Hearing Sciences [Book]. Plural Publishing, Inc. https://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=2013261&site=ehost-live

[63] Selcuk, S. (2017). Predictive maintenance, its implementation and latest trends. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 231(9), 1670–1679. https://doi.org/10.1177/0954405415601640

[64] Serradilla, O., Zugasti, E., Rodriguez, J., & Zurutuza, U. (2022a). Deep learning models for predictive maintenance: a survey, comparison, challenges and prospects. Applied Intelligence, 52(10), 10934–10964. https://doi.org/10.1007/s10489-021-03004-y

[65] Sirkin, & Harold. Why Advanced Manufacturing Will Boost Productivity.

[66] Sorjamaa, A., Hao, J., Reyhani, N., Ji, Y., & Lendasse, A. (2007). Methodology for long-term prediction of time series. Neurocomputing, 70(16-18), 2861-2869.

[67] Staufen.. (15. Oktober, 2019). Welche der folgenden Predictive-Maintenance-Anwendungen nutzen Sie bereits? [Graph]. In Statista. Zugriff am 29. Januar 2023, von https://de.statista.com/statistik/daten/studie/1078451/umfrage/nutzung-von-predictive-maintenance-anwendungen-in-deutschland/

[68] Stich, V., Henke, M., & Heller, T. (2019). Smart Maintenance – Der Weg vom Status quo zur Zielvision - acatech. acatech. https://www.acatech.de/publikation/smart-maintenance-der-weg-vom-status-quo-zur-zielvision/

[69] Streim, A., & Meiecke, C. (Mo., 2020, November 16). Corona treibt Digitalisierung voran – aber nicht alle Unternehmen können mithalten. *Bitkom E.V.* https://www.bitkom.org/Presse/Presseinformation/Corona-treibt-Digitalisierung-voran-aber-nicht-alle-Unternehmen-koennen-mithalten

[70] Suradhaniwar, S., Kar, S., Durbha, S. S., & Jagarlapudi, A. (2021). Time series forecasting of univariate agrometeorological data: a comparative performance evaluation via one-step and multi-step ahead forecasting strategies. Sensors, 21(7), 2430.

[71] Teoh, Y. K., Gill, S. S., & Parlikad, A. K. (2021). IoT and Fog Computing based Predictive Maintenance Model for Effective Asset Management in Industry 4.0 using Machine Learning. IEEE Internet of Things Journal, 1. https://doi.org/10.1109/JIOT.2021.3050441

[72] Traini, E., Bruno, G., & Lombardi, F. (2021). Tool condition monitoring framework for predictive maintenance: a case study on milling process. International Journal of Production Research, 59(23), 7179–7193. https://doi.org/10.1080/00207543.2020.1836419

[73] Uakhitova, A. B. (2022). Electricity Consumption Forecast Based on Neural Networks. Mathematical Models and Computer Simulations, 14(5), 863–874. https://doi.org/10.1134/S2070048222050167

[74] Waterer, F. (2012). 7 Predictive Maintenance Technologies That Enhance Power Equipment Reliability White paper | Schneider Electric. https://www.se.com/ww/en/download/document/998-2095-01-16-12AR0_EN/

[75] Willamos Silva, & Miriam A. M. Capretz (2019). Assets Predictive Maintenance Using Convolutional Neural Networks. 2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD). https://www.semanticscholar.org/paper/Assets-Predictive-Maintenance-Using-Convolutional-Silva-Capretz/f7e2fa736d2862ebad3dfa9b9978db1e58b5898e

[76] Wirth, R. & Hipp, Jochen. (2000). CRISP-DM: Towards a standard process model for data mining. Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining.

[77] Xu, Q., Zhang, M., Gu, Z., & Pan, G. (2019). Overfitting remedy by sparsifying regularization on fully-connected layers of CNNs. Neurocomputing, 328, 69-74. https://doi.org/https://doi.org/10.1016/j.neucom.2018.03.080

[78] Yu, Y., Hu, C., Si, X., Zheng, J., & Zhang, J. (2020). Averaged Bi-LSTM networks for RUL prognostics with non-life-cycle labeled dataset. Neurocomputing, 402, 134–147. https://doi.org/10.1016/j.neucom.2020.03.041

[79] Zhang, W., Yang, D., & Wang, H. (2019). Data-Driven Methods for Predictive Maintenance of Industrial Equipment: A Survey. IEEE Systems Journal, 13(3), 2213–2227. https://doi.org/10.1109/JSYST.2019.2905565

[80] Zonta, T., Da Costa, C. A., Da Rosa Righi, R., Lima, M. J. de, Da Trindade, E. S., & Li, G. P. (2020). Predictive maintenance in the Industry 4.0: A systematic literature review. Computers & Industrial Engineering, 150, 106889. https://doi.org/10.1016/j.cie.2020.106889

Declaration of Authorship

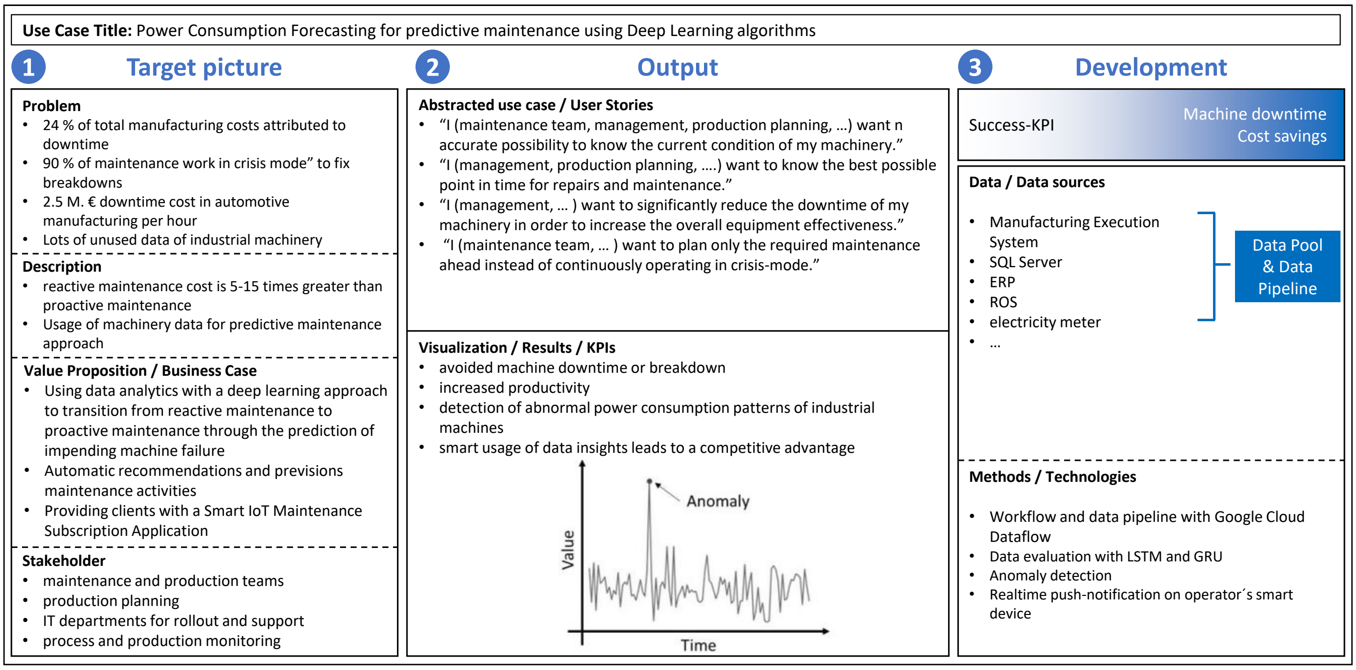
I hereby declare that the thesis submitted is my own unaided work. All direct or indirect sources used are acknowledged as references.

I am aware that the thesis in digital form can be examined for the use of unauthorized aid and in order to determine whether the thesis as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future theses submitted. Further rights of reproduction and usage, however, are not granted here.

This paper was not previously presented to another examination board and has not been published.

Appendix A

Business Understanding: Detailed version of our business model canvas



Appendix B

Exemplary embedding of our power consumption forecasting model in a cloud environment using the example of Google (Google Cloud, 2023)