CHALMERS
UNIVERSITY OF TECHNOLOGY

Investigation of machine learning approaches in process industry

Master's thesis in Innovative and Sustainable Chemical Engineering

ARSLAN FAROOQI

Department of Space, Earth and Environment
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

Master's thesis 2020

Investigation of machine learning approaches in process industry

ARSLAN FAROOQI

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Space, Earth and Environment
Division of Energy Technology
Chalmers University of Technology
Gothenburg, Sweden 2020

Investigation of Machine Learning Approaches in Process Industry
ARSLAN FAROOQI

© ARSLAN FAROOQI, 2020.

Supervisors: Christian Langner, Department of Space, Earth and Environment
Dr. Elin Svensson, Chalmers Industriteknik
Examiner: Professor Simon Harvey, Department of Space, Earth and Environment

Master's Thesis 2020:NN
Department of Space, Earth and Environment
Division of Energy Technology
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2020

Investigation of Machine Learning Approaches in Process Industry
ARSLAN FAROOQI
Department of Space, Earth and Environment
Chalmers University of Technology

Abstract

Today, industry is in continuous development, with digitalisation occurring at an increasing rate. *Industry 4.0* and *Internet of Things* are two common expression which both concern the development of industry with focus on artificial intelligence (AI). Studies have shown the benefits of applying various AI techniques in industry, e.g. more accurate fault diagnostics and better estimation of remaining useful life of process equipment.

The main aim of this master's thesis project was to investigate and apply machine learning methods as solutions to challenges identified in the process industry, and evaluate the outcome.

A literature review was first conducted to gain insight in the process industry and its current issues for which machine learning techniques could be applied. Thereafter, a study visit to Södra Cell's market pulp mill in Mönsterås was carried out, during which a case study suitable for data-driven modeling was identified. The case study involved one of the cooling towers at the mill, as well as process units in close proximity to the cooling tower, for which input and output variables were defined. The plant operators suspected fouling to negatively affect the performance of the cooling tower, and asked if the system could be modeled to identify the fouling effect. This hypothesis was investigated by modeling the system with two neural network architectures; Multilayer Perceptrons (MLP) and Long Short-Term Memory (LSTM), the latter being a type of recurrent neural network (RNN). The neural networks were modeled based on data extracted from four years with a time resolution of one hour. Comparing the results of the MLP networks with the LSTM networks led to the identification of recurrency, which in this case referred to the fouling effect.

Out of all output variables, conductivity was set as the target variable, or the variable of extra interest, as it was assumed to directly correlate with the amount of fouling occurring in the cooling tower system. As fouling occurs, there is an increase of metal ions in the recirculating cooling stream, leading to an increase in the measured conductivity.

The results showed that all LSTM networks, except for one, obtained better model accuracy than the MLP networks. The best MLP network yielded a value of Mean Squared Error (MSE) $MSE_{MLP2} = 0.003067$ while the best LSTM network, LSTM 7, yielded $MSE_{LSTM7} = 0.001335$. Furthermore, all LSTM networks, regardless of overall performance, modeled the conductivity output better than the MLP networks. The results give a clear indication that there is some recurrency present in the modeled system, which confirms the plant operators' hypothesis of fouling oc-

curing in the system. The statistical model yielded in this thesis work could then be used as groundwork for future projects. By accurately predicting the performance of the cooling tower system, investment decisions and optimisation of operational conditions could be carried out.

Keywords: Industry 4.0, Internet of Things, Multilayer Perceptrons, Neural Networks, Recurrent Neural Networks, LSTM, Machine Learning, Process Industry, System modeling, Pulp Mill.

Acknowledgements

During the course of this master's thesis, there have been several people, without whom this project would have been impossible to accomplish. First of all, I would like to thank my supervisor Christian Langner. His interest in this topic, as well as his knowledge and eagerness to discuss ideas, have been invaluable to me.

I would also like to thank my supervisor, Dr. Elin Svensson. With her knowledge and ability to keep up-to-date with recent literature, she steered me in the right direction whenever I was getting stuck or off-track. I also thank my examiner, Professor Simon Harvey. With his immense experience in academia, I could get insightful and relevant feedback to improve.

I also extend my gratitude to the people at Södra Cell Värö, Johan Isaksson and Andreas Darnell. During a very productive meeting with Andreas, several interesting ideas for the project were discussed. I also thank Magnus Tyrberg and the people at Södra Cell Mönsterås, for taking their time to meet with us and give us the opportunity to work with their pulp mill.

Arslan Farooqi, Gothenburg, May 2020

Contents

List of Figures

List of Tables

1
Introduction

The world's economy is growing faster than ever, and with it grows the demand for energy. This demand is mainly satisfied through conversion of fossil fuels. Therefore, the increase in energy use comes at the expense of the environment, which in turn impedes the climate goal of keeping the global average temperature well below two degrees, set during the Paris Agreement [1]. Reducing the energy consumption worldwide is therefore of great importance, which is why the EU has established the Energy Efficiency Directive [2]. This directive was revised in 2018, and member states of the European Union have now been commissioned to improve energy efficiency by at least 32.5 % until 2030, compared to 2007 [3]. This improvement in energy efficiency is expressed as reductions in consumption of primary energy and final energy [2].

In Sweden, national targets have been formulated which take economic development into consideration. The government has commissioned the Swedish Energy Agency to improve the intensity of Swedish energy consumption by 50 % until 2030, compared to 2005 [4]. In this context, energy intensity is measured as primary energy use per GDP [4]. Several different areas to be improved have been identified, one of which is the industrial sector. In 2018, Swedish industry accounted for approximately 38 % of the final energy use in Sweden, 141 TWh out of 373 TWh [5]. Based on data from 2015, the pulp and paper sector accounted for 52 % of industry's final energy use [6], which equals approximately 20 % of Sweden's final energy consumption. This indicates the importance of improving energy efficiency in the pulp and paper industry.

1.1 Background

Södra Cell Mönsterås is one of the largest pulp mills in Sweden, with an annual production of 750 000 air-dried tons of pulp [7]. In addition, the mill also delivers district heating to the local community, exports green electricity, and produces tall oil which can be used as a feedstock for biodiesel production [7]. In response to the Swedish Energy Agency, Södra Cell Mönsterås has launched an initiative in which the pulp mill plans to reduce their consumption of heat and electricity [8]. In addition, the mill also aims to produce excess electricity and increase production efficiency in order to reduce their consumption of chemicals [8].

One possibility for Södra Cell Mönsterås, among others, to pursue the aims set

in the above-mentioned initiative is to investigate if current operational strategies can be optimised with respect to these aims. In this context, it is important to take variations in operating conditions into account. The mill and all its processes must be able to handle system variations with varying time-scale and magnitude. Especially seasonal variations are of importance in Sweden, as the surrounding environment changes significantly depending on the season. Besides seasonal variations, variations on shorter time-scales and disturbances need to be considered. For example if disturbances in operating conditions are not handled properly, they may propagate and result in increased demands of energy and chemicals.

Previous work has been carried out with regard to Södra Cell Mönsterås with a special focus on the management of seasonal variations. In 2017, Nihlmark and Mahmoud conducted a detailed inventory of operating data for the secondary heating system of Södra Cell Mönsterås, as the mill was lacking data in this area [9]. Additionally, a Pinch analysis was conducted to quantify the potential for energy savings within the mill's secondary heating system. In 2018, Bokinge and Erlandsson developed different retrofit proposals for a subsystem of the mill's secondary heating system based on the results of Nihlmark and Mahmoud. The performance of the different retrofit proposals was evaluated with respect to seasonal variations using a computational tool developed by Bokinge and Erlandsson [10]. Furthermore, Persson and Berntsson investigated seasonal variations and short-term variations and their influence on the opportunities for energy savings in a pulp mill. They concluded that compared to steam savings targets generated using annual average process data values, the inclusion of monthly average values in their models reduced the potential for steam savings by 15 % [11]. In an additional study, Persson and Berntsson showed that including average values of even shorter time-periods (i.e. average values for 10-minute intervals) reduced the potential for steam savings even further [12].

The combined results of the previous works indicate that variations in operating data should preferably be considered when investigating opportunities for improving energy efficiency of industrial energy systems. However, traditional methodologies for industrial energy systems improvements, such as pinch-based approaches, have been developed primarily for steady-state values. Therefore, considering variations in operating data can be quite tedious since e.g., trial and error approaches may be necessary. In order to evaluate how energy efficiency of (industrial) energy systems is affected by varying operating data usually mathematical modelling is required. These mathematical models can be evaluated and analyzed, e.g. by means of simulation software. However, the systems involved in process industries are large and complex and so are the mathematical models describing these systems. Consequently, the mathematical evaluation of these models can be computationally burdensome and may require substantial computational capacity. Additionally, inaccuracies of mathematical models may cause discrepancies between measured data and the results obtained by these mathematical models. Due to these discrepancies, there is a risk that various industry systems, which are built upon the mathematical models, will perform poorly. Thus, there has risen an interest in investigating these

problems through data-driven modelling, by the means of artificial intelligence (AI) and one of its sub-groups, machine learning.

The Merriam-Webster Dictionary defines machine learning as:

> *"the process by which a computer is able to improve its own performance (as in analyzing image files) by continuously incorporating new data into an existing statistical model"*[13]

To clarify the Merriam-Webster Dictionary's definition of machine learning, a machine learning algorithm generates a statistical model. The machine learning algorithm, or technique, continuously improves the performance of the statistical model, i.e. increasing the accuracy, by incorporating new and more data. Thus, the algorithm can produce a highly accurate statistical model, as long as there is sufficient data available. Consequently, as the accuracy of the machine learning models are dependant on data, they cannot be applied in cases where there is a lack of reliable data.

As mentioned, machine learning can be utilised to generate statistical models with high accuracy if sufficient amount of data is available. Commonly in industry, numerous amounts of sensors are used to measure data, which is then fed to control and monitoring systems. Subsequently, this data is logged and rarely used for any further purpose. On the other side, the amount of available data strongly encouraged approaches based on machine learning. Due to the algorithm's ability to improve the statistical model it produces, paired with excessive amount of data available, the statistical model can become very accurate. The fundamental difference between the traditional approach and the machine learning approach is that the traditional approach is equation-based while the machine learning approach produces models based solely on data.

Another advantage with the machine learning approach is that it uses available data, and requires no changes to the process systems or any further investments. The lack of costs associated with this approach should motivate industry to investigate machine learning methods for handling problems in process industries.

1.1.1 The AI Potential

Many industrial sectors are currently being digitalised at an increasing rate. *Industry 4.0*, the fourth industrial revolution, and *Internet of Things* are two expressions often mentioned, which both refer to the development of industry with AI as a centre point.

Several studies have been carried out where the benefits of AI in process industry have been demonstrated. Ragab et al. utilised machine learning, a subgroup of AI, for fault diagnostics in industrial chemical processes [14]. The authors used the process of Logical Analysis of Data (LAD), a machine learning technique, to build a decision model with the purpose of diagnosing faults during operation [14]. This was applied to a case study using data from a real black liquor recovery boiler in a

pulp mill, with results showing good accuracy of fault diagnostics [14].

Furthermore, Costello et al. constructed a machine learning model in order to accurately determine the remaining useful life of gas circulator units [15]. By using a data set containing historical operational data of a real gas circulator unit, the constructed model was able to measure the health state of operating gas circulator units in real-time [15].

In an additional study, Sainlez and Heyen compared the performance of several machine learning techniques with the performance of multilinear regression models, in predicting atmospheric pollutants in the exhaust gas of the recovery boiler in a Kraft pulp mill [16]. The results showed that some of the machine learning techniques were substantially more accurate than multilinear regression models, in predicting the NO_x emissions of the recovery boiler.

Furthermore, a comprehensive system analysis was carried out by PiiA (Processindustriell IT & Automation) together with Blue Institute. In this study, the authors investigated how the implementation of AI can impact industry [17]. The authors established the benefits of AI development in industry by presenting several currently ongoing projects. In the report, the authors also analysed the market potential of AI. They forecasted that the market for AI will grow by up to 40% per year, due to the benefits achieved through AI solutions [17].

1.2 Aim

The main focus of this project is to approach challenges in process industry from a data-driven perspective. The overall aim can be defined as:

> *Investigate machine learning methods as solutions to challenges identified in process industry, and demonstrate the validity of these methods, by the means of data-driven modelling.*

The project focused on identifying case studies at Södra Cell Mönsterås which are suitable for the application of machine learning methods, and demonstrating the validity of the applied methods. The specific goals of the project were as follows:

- Literature review to gain insight in process industry challenges for which machine learning approaches can be applied.
- Identification of case studies at Södra Cell's pulp mill in Mönsterås, suitable for the application of machine learning methods.
- Through literature review, map machine learning methods to determine which approaches can be applied to the case studies identified in the previous step.
- Modelling of identified (sub)systems included in the case studies, by means of mapped methods and utilisation of available data.
- Analysis and validation of results in order to further improve the statistical models.

By applying machine learning methodologies to challenges in the process industry,

statistical models with good accuracy can be produced. In terms of model accuracy, these statistical models can perform better than equation-based models due to the ability of machine learning models to capture previously unknown correlations and dependencies. The derived machine learning models can then be used as groundwork for future work in improvements of various areas in process industry, which subsequently can lead to improvements in energy efficiency and reduced material intensity. Examples of this could be improved controllability or earlier detection of malfunctioning such as fouling of heat exchangers.

2
Theory

This chapter aims to introduce the reader to the knowledge required to understand and interpret the results of this work. First, the reader will be presented with a general description of machine learning. Following this, a detailed description of the machine learning techniques used in this project will be presented.

2.1 Machine Learning

The various techniques within machine learning can be grouped into categories, which are presented in Table 2.1 together with the most common applications for each category.

Table 2.1: Summary of the categories within machine learning.

Machine Learning Category	**Typical applications**
Supervised Learning	Regression modeling for descriptive and predictive analysis
Unsupervised Learning	Pre-processing and post-processing of data
Reinforced Learning	Robotics

In this thesis, machine learning algorithms based on supervised learning have been utilised in order to generate statistical models of (sub)systems of process industry plants with high modelling accuracy.

2.2 Supervised Learning

As the name indicates, supervised learning is a subset within machine learning which refers to all machine learning techniques where the training of the algorithms is supervised. Supervised learning or training implies the availability of data which the machine learning algorithm can utilise to determine the accuracy of a statistical model. By comparing the model output to the true output data, the algorithm can improve the statistical model and minimize the error between the two outputs.

Within supervised learning, there are several subgroups such as decision trees (DT), support vector machines (SVM) and artificial neural networks (ANN). In this master thesis project, ANNs were the main focus and are explained further in 2.2.1.

2.2.1 Artificial Neural Networks

ANNs are systems that loosely resemble the network of neurons in animal brains. They consist of an input layer, in which data is fed to the network. This data is then passed through hidden layers of neurons, and ultimately to the output layer, in which an output is acquired. During training, the model output is compared to the true output, often called label, and the ANN can then improve the accuracy of the statistical model, based on the difference between the output layer and the label, by aiming to minimise the error between the two. Within ANNs, there are several various modelling approaches, such as Multilayer Perceptrons (MLP) and Recurrent Neural Networks (RNN), which both will be described in detail.

2.2.1.1 Multilayer Perceptrons

A Multilayer Perceptron network is the simplest ANN and is illustrated in Figure 2.1.

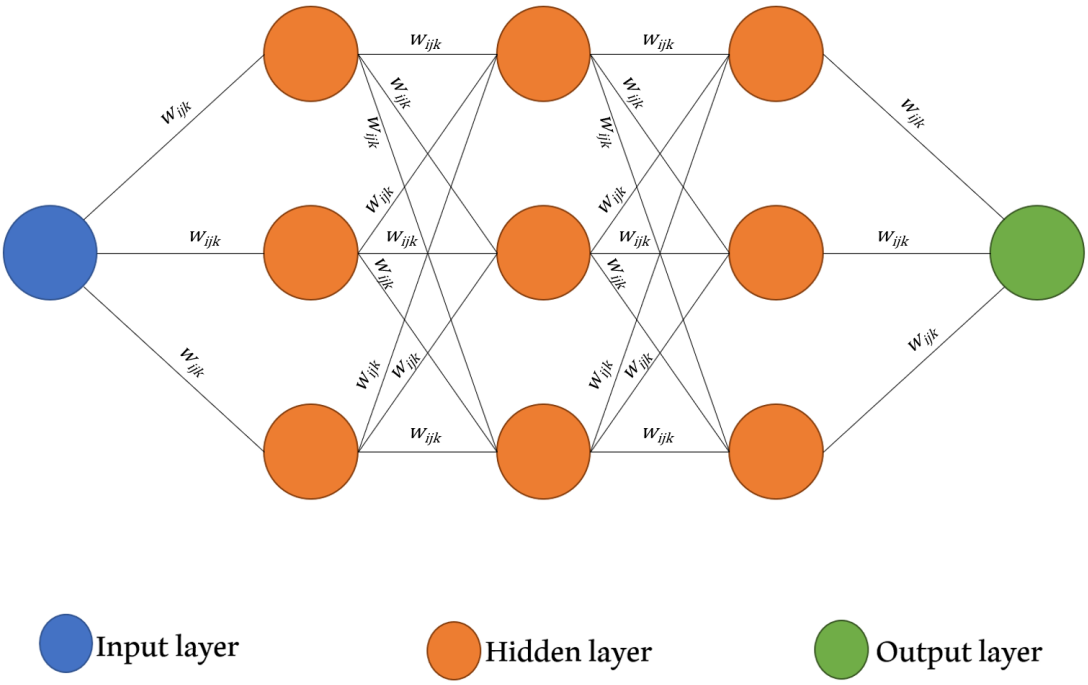

Figure 2.1: Illustrative figure of how a typical MLP is connected.

In Figure 2.1, the output values of each neuron propagate forward to the next layer, multiplied with the weight w_{ijk}. The output value is generated by applying an activation function to the input value of a neuron. There are several activation functions, such as the sigmoid function, the *tanh* function as well as the rectified linear unit (ReLU) function. In this project, the only considered activiation function was ReLU. During training, the machine learning algorithm improves the accuracy of the statistical model it produces, by changing the values of these weights in order to minimise the difference between model output and the true output which can be described by means of a loss function (see Subsubsection 2.2.1.2). The complexity of

these kinds of statistical models can be increased by increasing the amount of neurons per layer, as well as increasing the total amount of hidden layers. Generally, deep learning refers to neural networks with more than three hidden layers. This added depth allows the complexity of process industry systems to be retained in the statistical model, given that there is sufficient data available for training.

To illustrate the mathematical operations happening in an MLP, Figure 2.2 shows two hidden layers, with two neurons each, of an MLP network.

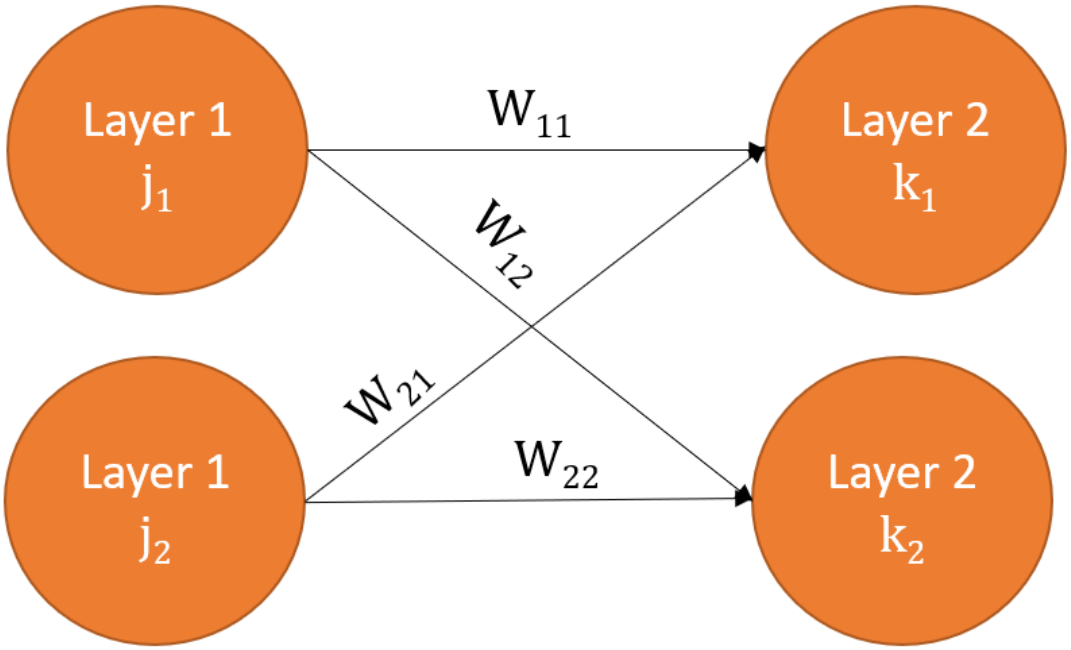

Figure 2.2: Two hidden layers of a MLP network.

The values j_1 and j_2 are the outputs of the neurons in the first layer, while k_1 and k_2 are the outputs of the neurons in the second layer. The outputs of the second layer are then defined by,

$$\begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = f_a \left(\begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} \begin{bmatrix} j_1 \\ j_2 \end{bmatrix} \right) \tag{2.1}$$

where f_a is the chosen activation function (e.g. ReLu).

2.2.1.2 Loss functions

For artificial neural networks, the loss function is the function which the machine learning algorithm minimises during its training. There are several metrics of loss, some of which are presented in Table 2.2 together with the problems they are typically used for. In this project, Mean Square Error (MSE) was used as a loss metric while e.g. crossentropy loss metrics are generally used for various classification problems, such as image recognition.

Table 2.2: Various loss metrics available for artificial neural networks.

Loss metric	**Common applications**
Mean Square Error (MSE)	Regression modeling
Mean Absolute Error (MAE)	Regression modeling, outliers have less impact
Binary Crossentropy	Multi-label classification
Categorial Crossentropy	Single-label classification

The MSE loss function is defined as:

$$L(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^{N} (y - \hat{y}_i)^2,$$

where y is the true output, $\hat{y}$ is the predicted model output and N is the total amount of observations, or data points. In contrast, the MAE loss function is defined as:

$$L(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^{N} |y - \hat{y}_i|$$

The main difference between MSE and MAE is that MSE squares the error, meanwhile MAE only accounts for the absolute difference. This means that for MSE larger model errors affect the loss function to a greater extent.

2.2.1.3 Recurrent Neural Networks

Recurrent neural networks (RNN), similar to MLP, is a type of ANN used for regression modeling. In contrast to MLP, RNN account for recurrency, meaning that the outcome at a given point is influenced (to some extent) by the outcome of previous points. This means that the order of data points in the data set is extremely important, as the data is processed sequentially, rather than in a randomised fashion, which can be the case for MLP. Figure 2.3 illustrates a simplified version of a RNN. There are also various types of RNNs. One such type is the long short-term memory (LSTM) network, which will be used in this project.

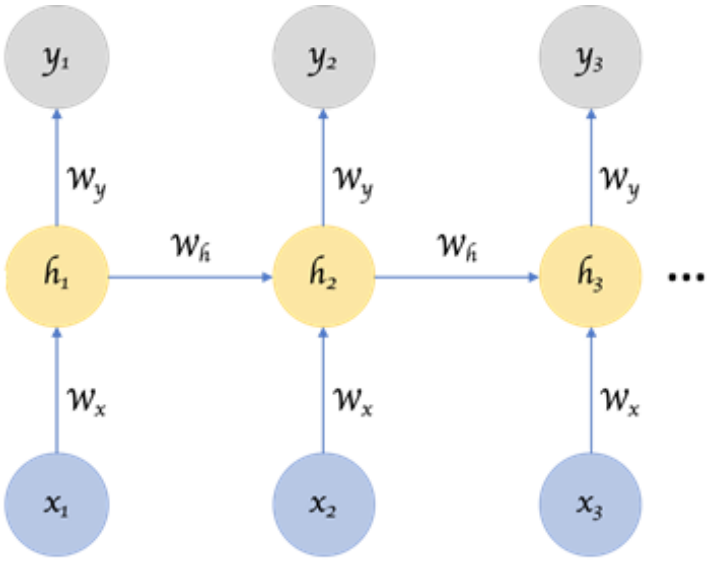

Figure 2.3: Simplified illustration of how a typical RNN is connected. [18]

In Figure 2.3, each set of x, h and y represent a typical MLP network, where x is the input layer, h are the set of hidden layers and y is the output layer, as shown in Figure 2.1. However, instead of processing the entire data set in one such network, as is the case with MLP networks, the data is divided in batches, called time steps, creating a time series.

As an example, a dataset consisting of six data points could be divided into time steps of three data points each. This would mean that the first time step included data points one through three, the second time step consisted of data points two through four, and so forth, until data point five is reached. At this point, the last two data points, data point five and data point six, would be omitted since a full batch size of three data points cannot be reached.

In Figure 2.3 each batch, or time step, is solved in a set of x, h and y, and some of the output is propagated forward to the next batch, through the weight factor w_h. This means that the first batch is processed in the set consisting of x_1, h_1 and y_1. The second batch would then be processed in the set consisting of x_2, h_2 and y_2, with the influence of the previous batch, through the weight factor w_h, and so forth.

RNNs are suitable for regression modeling of time series data, meaning data in which time is an important dimension. Thus, in this project, the terms time-dependancy and time-dependant phenomena both refer to the behaviour of recurrency, where the outcome of the previous data points affects the outcome of the current and following data points, which RNNs account for in the modeling.

2.2.1.4 Hyperparameters

In machine learning, hyperparameters are parameters whose values, or settings, are pre-defined, prior to the training, and do not change during the process. As an example, the number of hidden layers or the number of neurons per layer, are both hyperparameters for MLP networks and LSTM networks. In Table 2.3, the hyperparameters for MLP networks and LSTM networks are presented.

Table 2.3: Some of the available hyperparameters for MLP networks and LSTM networks.

Hyperparameter	MLP network	LSTM network
Number of hidden layers	Yes	Yes
Number of neurons per layer	Yes	Yes
Learning rate	Yes	Yes
Number of data points per time step	No	Yes

All hyperparameters presented in Table 2.3 have been described in Section 2.2.1.1, except for learning rate. This is done in the following. When the machine learning algorithm changes the weight factors of an MLP or RNN to minimise the loss function, the magnitude of changes to the weight factors is determined by the learning rate. A high value for the learning rate means that the weight factors are being modified in large increments, and a small value for the learning rate results in the weights being changed in small increments. Thus, the value of the learning rate can affect the training results drastically. High values for the learning rate can result in the algorithm minimising the loss function in a small amount of time. However, large learning rates can also result in oscillating or unstable behaviour of the model, if the steps taken are too large to accurately find the point where the loss is minimised. In contrast, small values of the learning rate often lead to the loss function being properly minimised, although at the cost of substantially longer training time.

2.3 Unsupervised learning

The second main category of machine learning, whose main constituents are various clustering algorithms, is unsupervised learning. These clustering algorithms are often used to identify certain commonalities of data points within data sets, which can subsequently be grouped in clusters. Compared to supervised learning previously explained by means of ANNs, unsupervised learning differs in that there are no labels which can be used to train the algorithms. Since there are no labels and thus no feedback, the algorithms operate by identifying attributes present within the dataset. Data with similar attributes is then grouped together. Generally, this process can take many forms, depending on the clustering algorithm used.

Unsupervised learning techniques are often used for their powerful data processing abilities, such as independant component analysis (ICA) and dimensionality

reduction.

2.4 Limitations of machine learning

There are, however, some general drawbacks with machine learning methodologies. When machine learning algorithms are trained, they excel at modeling and predicting outcomes in the range of the given data, with some extrapolation included. However, if the system for which the algorithm has been trained, is changed to a great degree, the algorithm will fail to anticipate the outcome. An example of this can be observed in a system of heat exchangers. If one or more heat exchangers are moved, or new heat exchangers are introduced to the system, the machine learning algorithm will not be able to predict the subsequent change in energy demand.

3
Methods

This chapter aims to describe the planning and execution of the thesis work, which was divided into two main parts. The first part relates to a study visit at Södra Cell Mönsterås, during which two case studies were defined. One case study was then included for further analysis. Subsequently, several machine learning methods were investigated and mapped with respect to their assumed applicability to model the chosen case study. The details of the identification and mapping are described in Section 3.1. Following this, the modelling of systems involved the identified process industry challenges, via the means of the mapped machine learning methods, was carried out. This part included filtering and processing of data from Södra Cell Mönsterås, as well as coding and using the identified machine learning methods to model the chosen case study. Finally, the results were analysed with respect to the achieved model accuracy. Using the results as feedback, the data pre-processing steps and hyperparameters were modified in an iterative fashion, to improve the model accuracy. This is described in detail in Section 3.2. The entire workflow is illustrated in Figure 3.1.

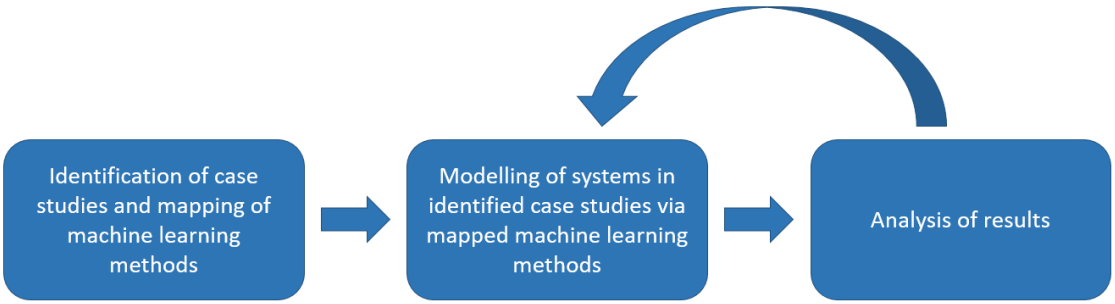

Figure 3.1: Illustration of the general workflow process of the master thesis.

3.1 Identification and mapping

This section aims to describe the details of work carried out in the first part of the project, which includes the study visit to Södra Cell Mönsterås and the case study defined there. Figure 3.2 illustrates all included parts and the work flow.

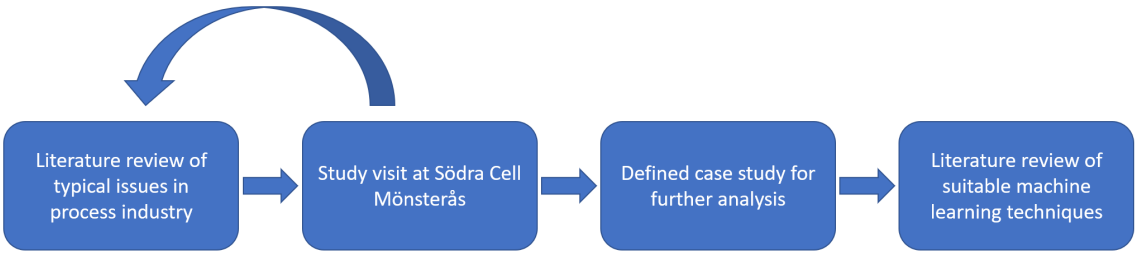

Figure 3.2: Flowchart of the work process in the first part of the project.

At an initial stage of the project, a thorough review of relevant literature was carried out. Through the literature review, a general overview on typical challenges in process industry was obtained. Furthermore, examples discussed in the literature were identified where machine learning approaches were applied in process industry. This information was then used as a base for the meetings conducted during the study visit at Södra Cell Mönsterås.

3.1.1 Study visit at Södra Cell Mönsterås

In the middle of February, a week-long study visit at Södra Cell Mönsterås was carried out. During the first two days, discussions and meetings with engineers and plant operators were conducted. In these meetings, two case studies were identified to be suitable for further consideration. During the selection process a relevant research question was defined for each case study and the availability of measurement equipment and historical data was investigated. With respect to the extent of this thesis project, it was decided to select one case study for further analysis. Following this, the system boundaries were defined by reviewing flow sheets of the relevant parts of the pulp mill. Additionally, detailed mapping of sensors and measurement devices was conducted. This step was of great importance as the identified sensors and measurement devices were used at a later stage in the project, for the extraction of data.

3.2 Data-driven modelling

The outcome of the first part of the project was a well defined case study involving a subsystem at the pulp mill Södra Cell Mönsterås, which was modeled using the identified machine learning methods. This process included the coding and training of the selected machine learning methods and the statistical models produced. As described previously in this thesis machine learning approaches are highly dependant on the available data. Thus, gathering, filtering and processing of data were all detrimental steps in ensuring good accuracy of the statistical models. An illustration of the work flow in this part of the project is presented in Figure 3.3.

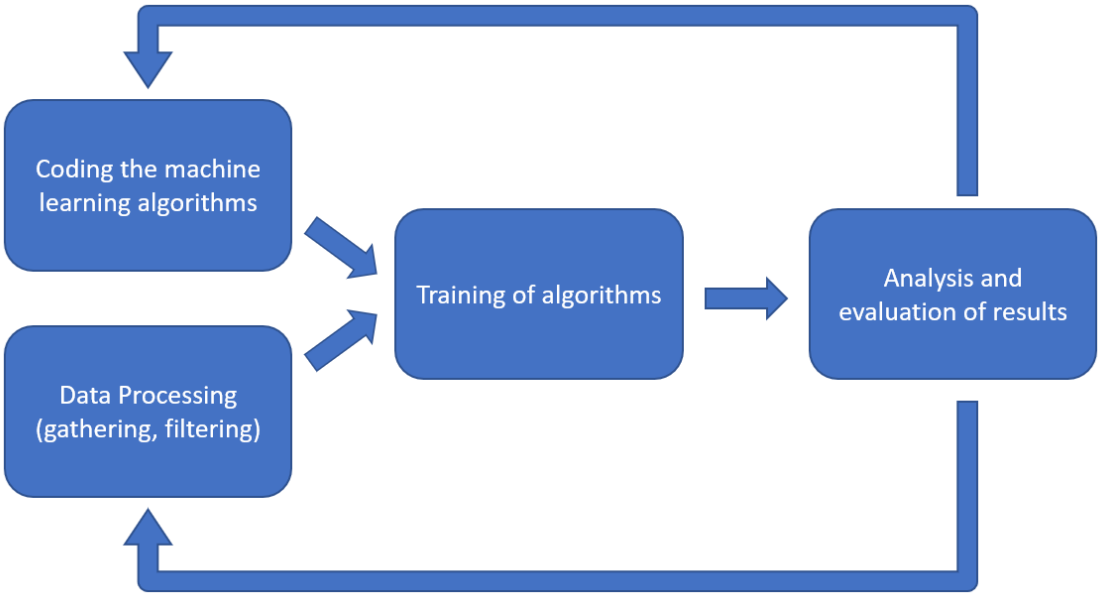

Figure 3.3: Work flow chart of the second part of the project.

All steps described in Figure 3.3 were done by utilizing different available packages of the programming language Python. In the following sections, the data-processing steps as well as coding of the machine learning algorithms are presented. The essential code is presented in Appendix A.

3.2.1 Processing of data

During the study visit, input and output variables were defined for the chosen case study. The data was then gathered with the help of the available monitoring interface in Microsoft Excel, allowing the data to be extracted for further processing. The data was extracted with a time resolution of one hour, from the beginning of 2016 to the beginning of 2020, exactly four years.

Once the data had been acquired, the first step was filtering. To ensure that the final statistical model has a good accuracy for typical operating conditions, the data was filtered to remove all periods of maintenance, as well as the transient behaviour occurring during start-up and shutdown. This is presented more in detail in Section 4.2.1. Furthermore, the data was normalised to be able to present all data by means of values between 0 and 1. The background for this normalisation step is further explained in Subsection 4.2.1. The code for filtering can be found in Appendix A.1 and the code for normalisation can be found in Appendix A.2

3.2.2 Construction and training of machine learning models.

With the data available in a filtered and processed format, the different machine learning methods presented in Chapter 2 were utilised to develop and train statistical models representing the case study. In this project, no generation of novel

machine learning algorithms was done. Two different architectures for ANNs were evaluated: MLP networks and LSTM networks. MLP networks are described in detail in Section 2.2.1.1 and LSTM networks are described in detail in Section 2.2.1.3.

Furthermore, the hyperparameters of the different network architectures were varied in an iterative procedure which is described in Subsection 3.2.3. A defined network (with fixed hyperparameters) was trained using the extracted data and the training was interrupted if no sufficient improvement of the loss metric was observed for several epochs but latest after 300 epochs. The code applied for the construction and training of MLP networks is presented in Appendix A.3. For the constructed LSTM networks, the code is presented in Appendix A.4.

3.2.3 Variation of hyperparameters

Based on the training results of each neural network, the hyperparameters were varied in order to investigate the influence of a specific hyperparameter on the model accuracy. The objective of this step was to achieve a high model accuracy. As previously explained, the accuracy was determined by the loss metric MSE, which was calculated continuously throughout the training of the models. Different hyperparameter settings were chosen and in Table 3.1, the chosen hyperparameter settings for the MLP networks are presented.

	Number of layers	**Neurons per layer**	**Loss metric**
MLP network 1	2	150	MSE
MLP network 2	4	150	MSE
MLP network 3	4	500	MSE

Table 3.1: Hyperparameter settings each of the MLP networks trained.

In Table 3.2 the chosen hyperparameter settings for the LSTM networks are presented .

	Number of layers	**Neurons per layer**	**Data points per time step**	**Learning rate**	**Loss metric**
LSTM network 1	2	50	3	0.001	MSE
LSTM network 2	2	100	3	0.001	MSE
LSTM network 3	2	150	3	0.001	MSE
LSTM network 4	2	250	3	0.001	MSE
LSTM network 5	2	500	3	0.001	MSE
LSTM network 6	4	50	3	0.001	MSE
LSTM network 7	6	150	50	0.0001	MSE
LSTM network 8	8	150	100	0.0001	MSE

Table 3.2: Hyperparameter settings for each of the LSTM networks trained.

4
Results

In this chapter, the results of the work, carried out as described in Chapter 3, are presented. The findings are presented in two sections. First, the results obtained during the study visit at Södra Cell Mönsterås are summarized. In the second section, the results from the model training process, described in Section 3.2, and the evaluation of these results are presented.

4.1 Study visit at Södra Cell Mönsterås

In the middle of February, a week-long study visit to Södra Cell Mönsterås was carried out. During this week, through conducted meetings and data gathering, two case studies were identified as described in Chapter 3. The first case study involved the plant's evaporator system, for which the plant has employed two different maintenance and cleaning routines. A research question was defined as engineers and operators at the pulp mill were interested in knowing if historical process data could be used to determine if one of the two cleaning routines is preferable with respect to performance indicators such as steam demand or usage of chemicals.

The second case study concerned one of the cooling towers at the plant. The plant operators suspected a decline in performance of the cooling tower over time, and were anticipating that some time-dependant phenomena such as fouling is the cause of the performance decline. Since the cooling tower in question provides cooling capacity for a condensing turbine running on steam produced in the recovery boiler of the pulp mill, the performance of the cooling tower explicitly affects the amount of generated electricity. It was therefore decided to investigate if historical data can be used to develop a statistical model of the cooling tower system to validate or rebut the suspicion of plant personnel.

After the identification of the two case studies, the case study concerning the cooling tower system was chosen to be the main focus of this master thesis project. In the following subsection, the entire cooling tower system and all relevant parameters are presented.

4.1.1 Cooling Tower Case Study

A schematic illustration of the cooling tower system considered in this project, is presented in Figure 4.1.

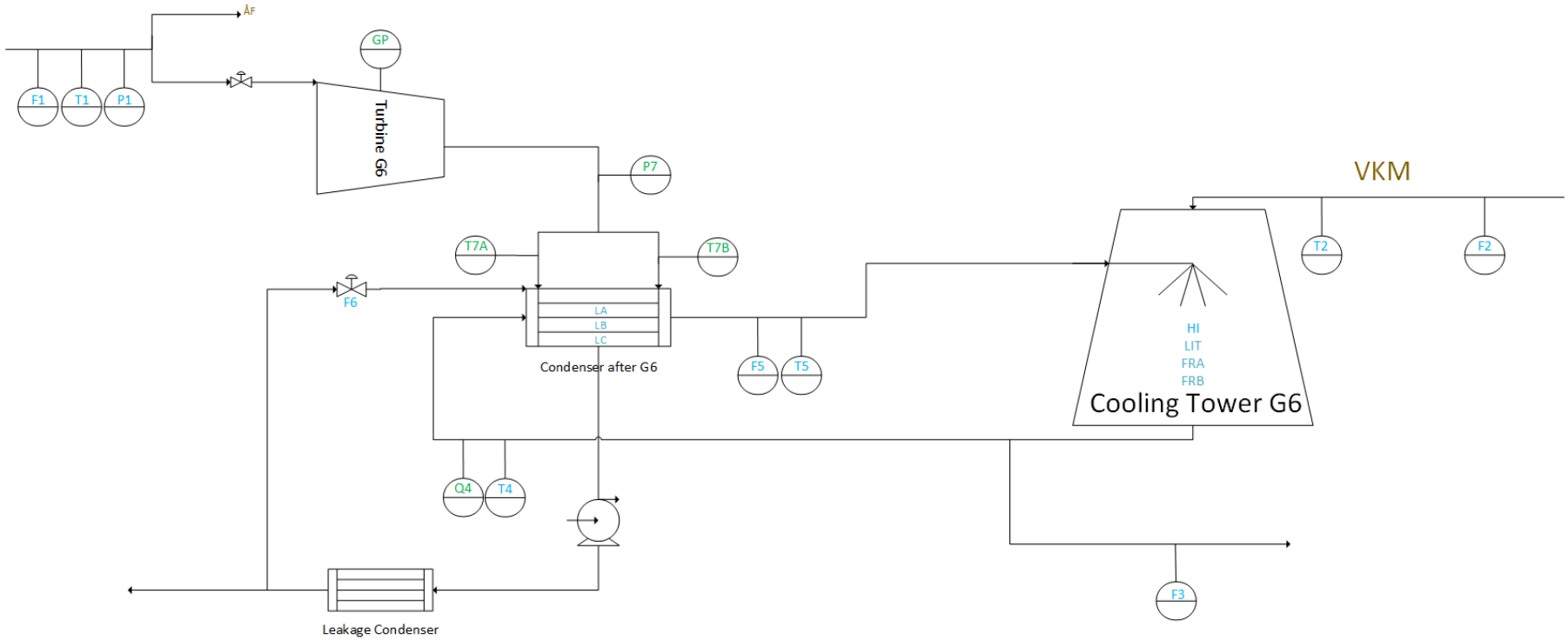

Figure 4.1: A schematic view of the system in the cooling tower case study.

In the schematic illustration, high pressure steam enters the turbine G6. After the expansion, the steam leaves the turbine at pressure and temperature set by the plant operators and controlled by vacuum pumps to be finally condensed in the condenser after turbine G6. The cooling medium utilised in the condenser is a stream of water taken from the cooling water pool of cooling tower G6. The water stream enters the condenser and leaves at higher temperatures. In order to maintain the cooling capacity, the water stream is recirculated back to the cooling water pool through spraying. This results in partial evaporation of the droplets into the ambient air which further results i a cooling effect. To make up for the water lost in the evaporation process, a level controller regulates an inlet stream of mechanically cleaned water, taken from a nearby river.

Furthermore, several sensors can be found in the cooling tower system. In Figure 4.1, the sensors are colour-coded according to how they were utilised in this project. A description of the colour-coding system can be found in Table 4.1.

Colour	**Utilisation**
Blue	Used as input variables for the machine learning models
Green	Used as output variables for the machine learning models

Table 4.1: Description of the colour-coding system.

Furthermore, a brief description of each sensor that was utilised, can be found in Table 4.2.

Sensor name	**Description**	**Measured unit**
Inputs		
F1	Calculated flow measurement at turbine inlet	kg/s
T1	Temperature measurement at turbine inlet	°C
P1	Pressure indicator at turbine inlet	MPa
LA	Level indicator in condenser after G6	mm
LB	Level indicator in condenser after G6	mm
LC	Level indicator in condenser after G6	mm
F5	Flow measurement of circulating cooling stream	kg/s
T2	Temperature measurement of mechanically cleaned water entering the cooling tower	°C
F2	Flow measurement of mechanically cleaned water entering the cooling tower	kg/s
F3	Flow measurement of mechanically cleaned water leaving the cooling tower	kg/s
T4	Temperature measurement of circulating cooling stream entering condenser after G6	°C
T5	Temperature measurement of circulating stream leaving condenser after G6	°C
HI	Ambient wet-bulb temperature measurement of environment surrounding cooling tower	°C
LC	Level indicator of reservoir in cooling tower	%
FRA	Speed measurement of cooling fan in cooling tower	RPM
FRB	Speed measurement of cooling fan in cooling tower	RPM
F6	Flow measurement of side stream recirculated back to condenser after G6	kg/s
Outputs		
GP	Measurement of power generation from generator G6	MW
P7	Pressure indicator of stream leaving turbine G6	kPa
T7A	Temperature measurement of stream entering condenser after G6	°C
T7B	Temperature measurement of stream entering condenser after G6	°C
Q4	Conductivity measurement of circulating cooling stream	mS/m

Table 4.2: A brief description of all included sensors in the project work.

As can be seen in Figure 4.1 and Table 4.2, a sensor measuring the conductivity of the circulating cooling water stream was utilised. This measurement is of extra interest as it might relate to the effect of fouling more than other parameters. The

hypothesis is that fouling or corrosion on the heat exchanger surfaces in the cooling tower system can lead to higher concentrations of metal ions in the circulating cooling water stream. This would then be observed as an increase in the conductivity measurement.

4.2 Model training and evaluation

In this section, the results obtained from model training and evaluation are presented. In a first step, historical process data was gathered from the sensors presented in Table 4.2, and preprocessed. This is described in Subsection 4.2.1. The pre-processed data was then utilised to train different statistical models by means of machine learning, and the results were evaluated (see Subsection 4.2.2). For all presented model results, the temperature output is presented as a mean of the two temperature measurement output variables, T7A and T7B.

4.2.1 Data Gathering and Preprocessing

As described previously, historical process data was gathered for the time period of four years, between 2016 and 2020. The data was acquired at a time resolution of one hour, resulting in about 35 000 data points. Once the data had been gathered, preprocessing was performed through the means of filtering. These preprocessing steps were done in order to eliminate all periods were the turbine was shut off, such as maintenance hours, as well as the transient behaviour occurring during shutdown and startup. Consequently, the power production of generator G6 was used as an indicator of the above mentioned events. This variable was used to filter the data, removing all data points according to the following criteria:

$$P(\text{Power Production}) < 5 \text{ MW}$$

In addition, ten additional data points were removed both before and after the periods for which $P < 5$ MW to account for the transient behaviour occurring during shutdown and startup. In Table 4.3, a detailed summary of the data gathering and preprocessing is presented.

Time period of dataset	2016-01-01, 10:00 to 2020-01-01, 09:00
Total amount of data points	35 064
data points after preprocessing	31 881
Removed data points	3183

Table 4.3: Short summary of the data gathering and preprocessing steps.

Finally, the entire dataset was normalised between 0 and 1, as described in Section 3.2.1. By normalising the data, the loss metric, which the model tries to minimise during training, becomes a more accurate representation of the model error.

4.2.2 Model training

In Section 4.1, a research question for the cooling tower case study was defined as that the plant operators suspected some time-dependant phenomena, which was then anticipated to result in a decline of the cooling tower performance, over time. To determine whether or not this was the case, several neural networks were trained and the results were compared and evaluated.

4.2.2.1 Multilayer Perceptron Networks

Initially, three different MLP networks were trained and the different sets of hyperparameters have been presented in Table 3.1. The hyperparameter settings as well as loss values obtained, for each MLP network, are presented in Table 4.4. As the chosen loss metric for all networks was MSE, a lower value means greater accuracy of the model.

	Number of layers	**Neurons per layer**	**Loss metric**	**Loss value**
MLP 1	2	150	MSE	0.004081
MLP 2	4	150	MSE	0.003067
MLP 3	4	500	MSE	0.005650

Table 4.4: Hyperparameter settings and loss values for each of the MLP networks trained.

One can observe that added complexity to the model, i.e. adding additional layers, yields better results in terms of model accuracy (compare MLP 1 and MLP 2). On the other hand, increasing the number of neurons from 150 to 500 did result in higher loss values (compare MLP 2 and MLP 3). In this context, it must be considered that training was interrupted after a maximum number of 300 epochs. Theoretically, a more complex model (as in additional neurons or additional hidden layers) should never yield a higher loss value if the model is given sufficient training time and assuming it is not stuck in a local optimum. However, the loss metric only gives an indication of the overall model accuracy, and does not describe how the model fits each individual output variable. In Figures 4.2 through 4.5, the predicted model outputs of the best performing MLP network (MLP 2) are compared to the true outputs of the dataset, where each figure illustrates each individual output variable (power production of turbine in Figure 4.2, pressure of steam after turbine in Figure 4.3, temperature of steam after turbine in Figure 4.4 and conductivity of circulating cooling water in Figure 4.5). The data is presented as a moving average over 200 data points, and the blue line represents the true output while the orange line represents the predicted model output. The output graphs of the remaining MLP networks (MLP 1 and MLP 3) are presented in Appendix B.1 and Appendix B.2.

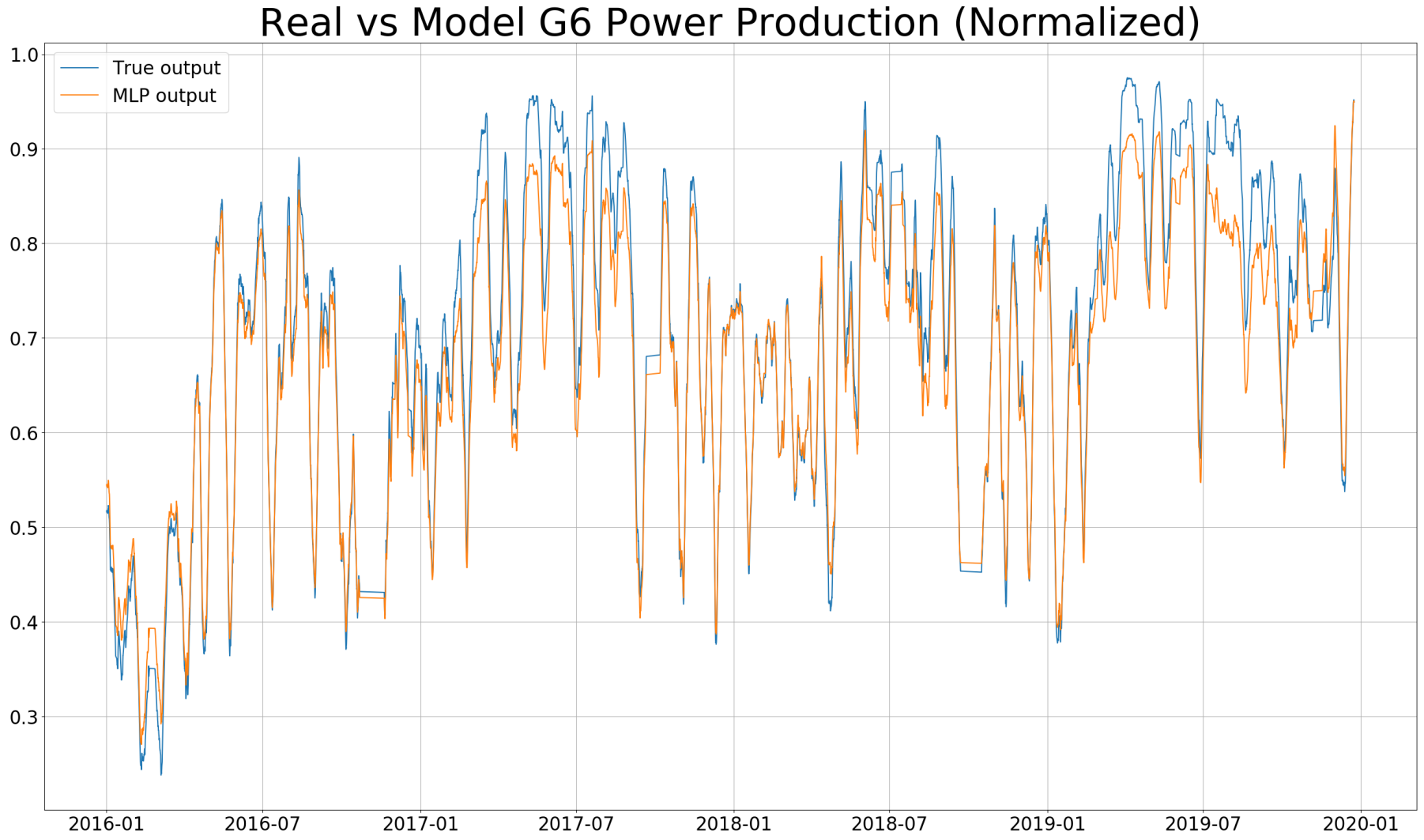

Figure 4.2: Power production output of MLP 2, compared to the true output.

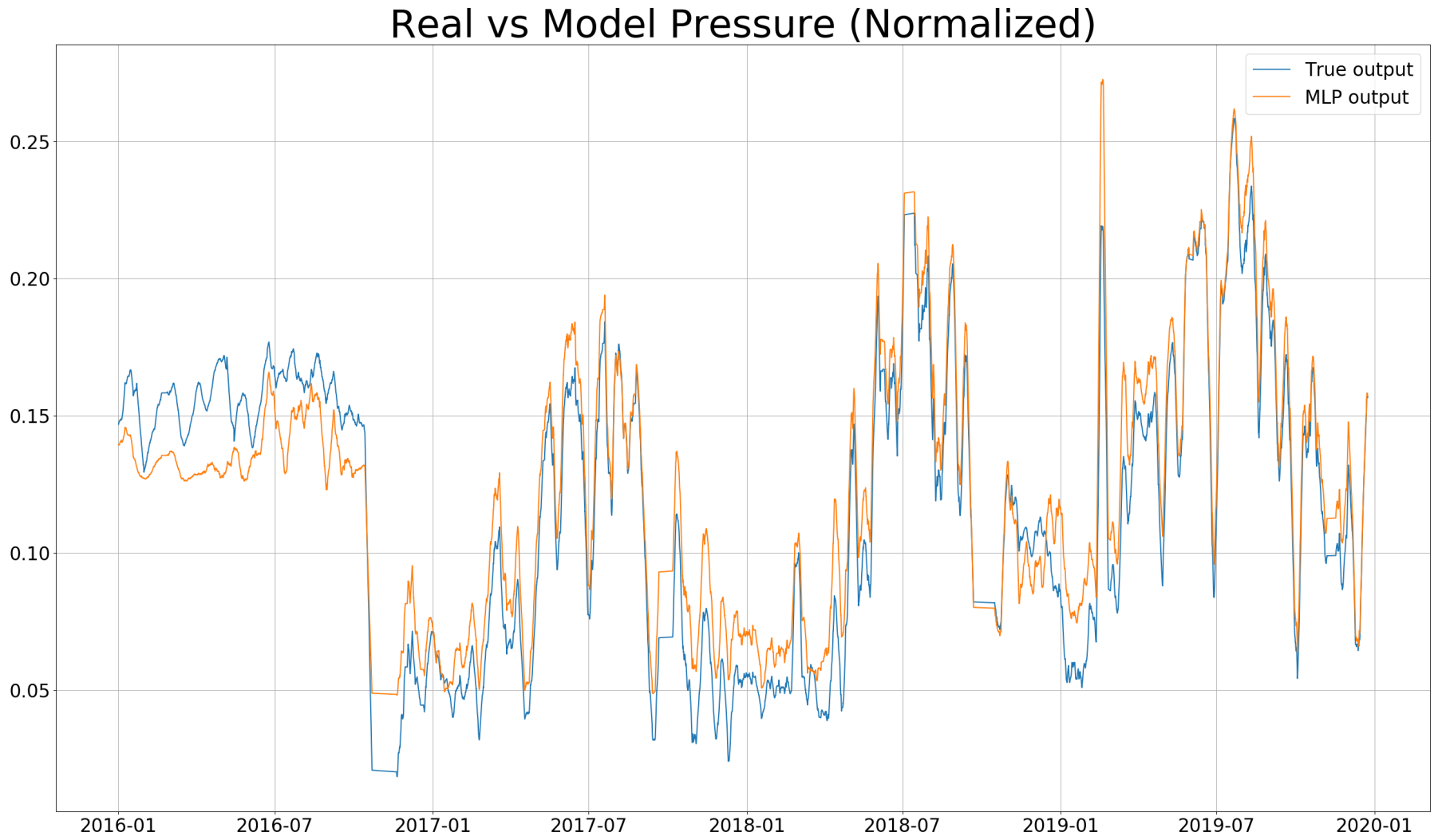

Figure 4.3: Pressure output of MLP 2, compared to the true output.

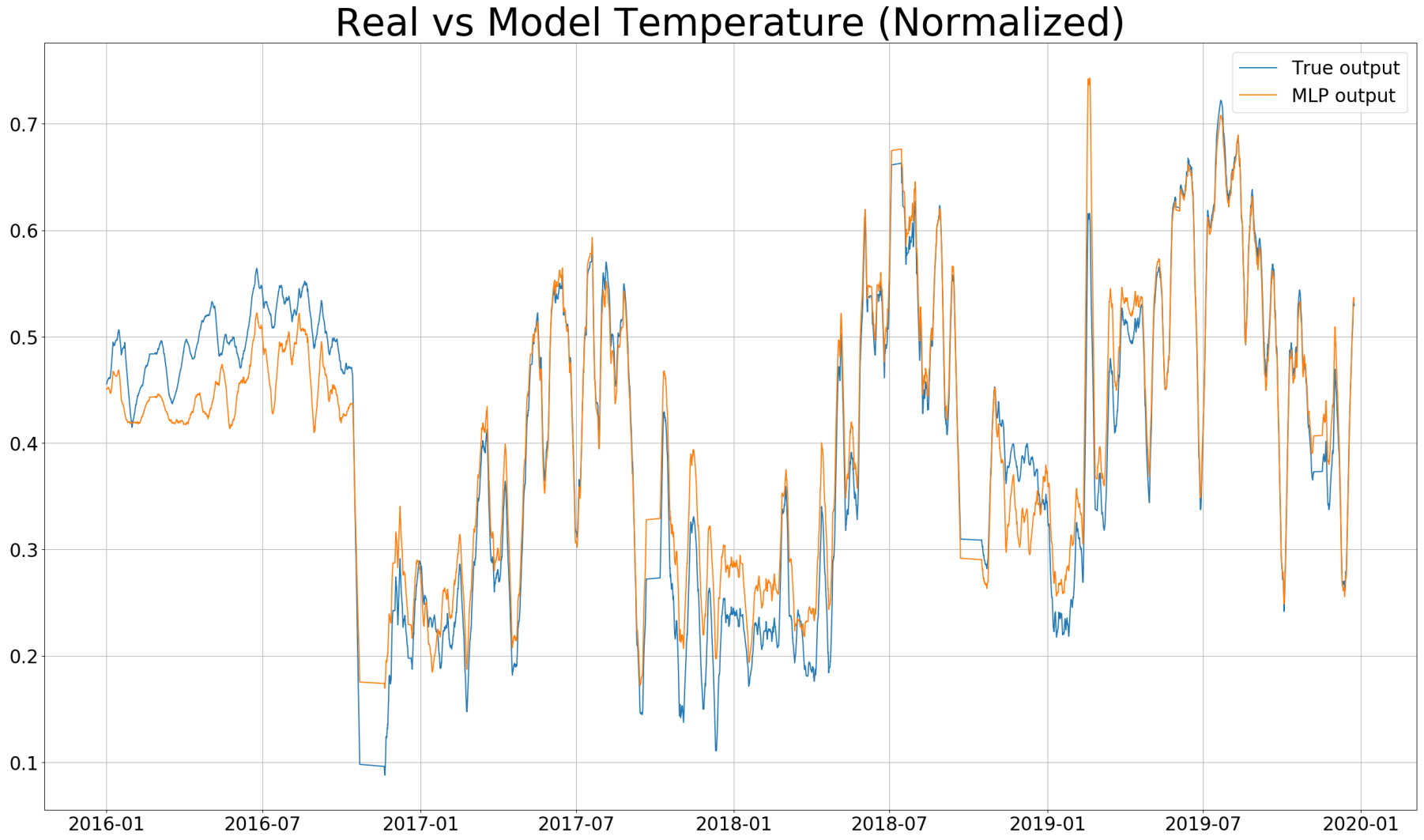

Figure 4.4: Temperature output of MLP 2, compared to the true output.

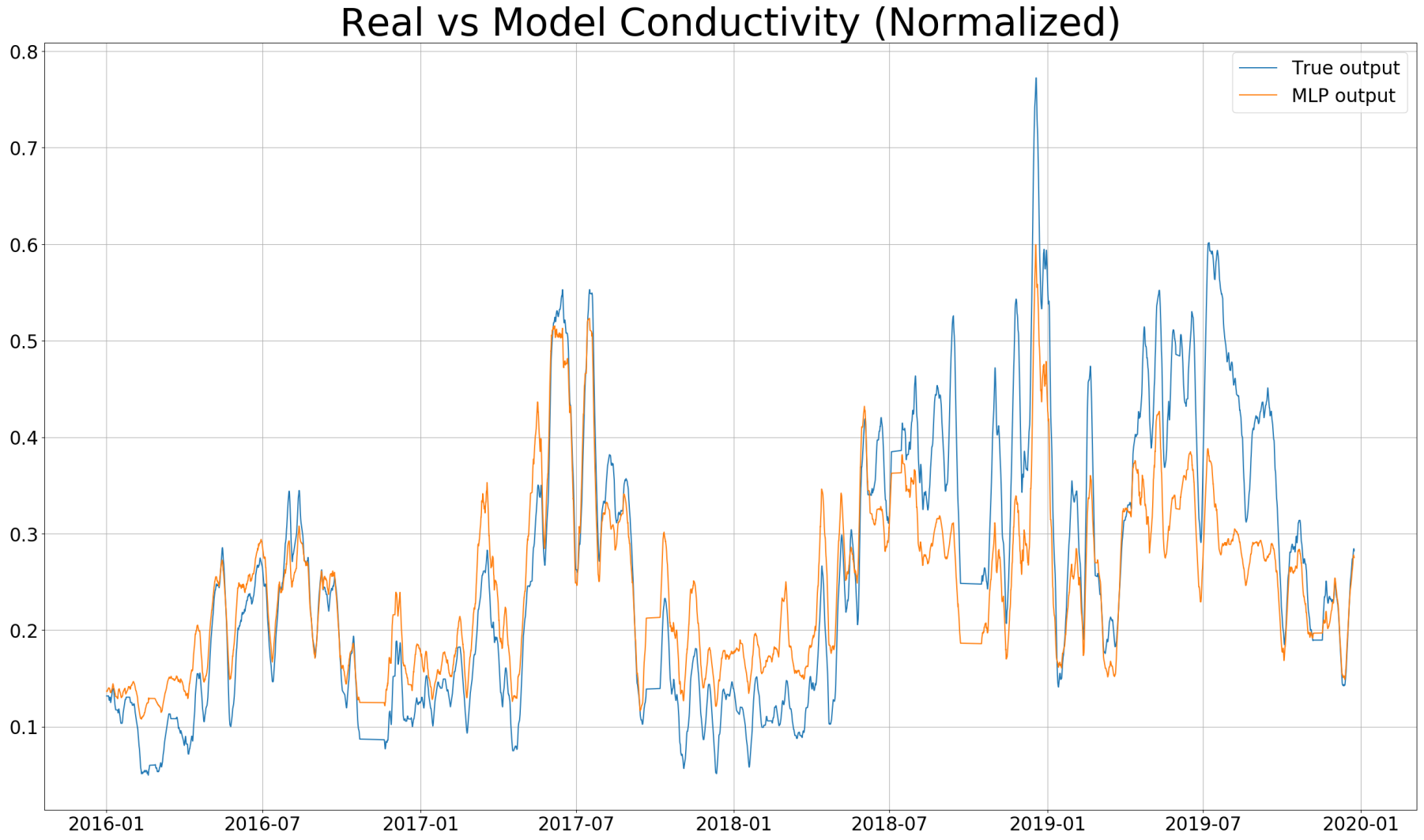

Figure 4.5: Conductivity output of MLP 2, compared to the true output.

Observing the graphs gives some indication of the accuracy of the model output, but small discrepancies between the variables are difficult to determine. In Table 4.5, the individual loss metrics for each output variable of all MLP networks are presented.

	Power production	**Pressure**	**Temperature**	**Conductivity**
MLP 1	0.001131	0.0006623	0.003199	0.01192
MLP 2	0.002063	0.0004330	0.001963	0.008826
MLP 3	0.0007746	0.0006143	0.003356	0.01964

Table 4.5: Individual MSE loss metric values for each output variable of all MLP networks.

All MLP networks predict conductivity substantially worse than the other output variables, as indicated by Table 4.5. The MSE of conductivity for all MLP networks is around one order of magnitude larger than for the other output variables.

While MLP 2 is the best performing network in terms of overall accuracy, the results show that this network has the poorest prediction of the power production, according to Table 4.5. However, MLP 2 predicts temperature and pressure substantially better than the other two MLP networks (MLP 1 and MLP 3). There is also a large discrepancy in conductivity prediction between MLP 2 and the other MLP networks. For conductivity, MLP 2 holds the lowest MSE of all MLP networks, as indicated by Table 4.5.

In Figure 4.5, the conductivity output of MLP 2 is compared to the true output. As indicated by the figure, the conductivity is predicted at an acceptable level until 2018-07 on the x-axis. After this, the prediction fails to a greater extent, indicating that there is something affecting the system which the MLP network does not account for.

By comparing Figures B.1 through B.4 and Figures B.5 through B.8 with Figures 4.2 through 4.5, as well as looking at the results in Table 4.4 and Table 4.5 it becomes evident that increasing the model complexity improved the model accuracy, but only to a certain extent. One can both graphically and numerically determine that the most complex MLP network, with four layers and 500 neurons per layer, performs the worst, as indicated by Table 4.4. However, the increased complexity, i.e. attempts of overfitting, did not yield improved results for the defined time of training. Thus, it can be concluded that there occurs some phenomena which is intrinsically hard to capture by means of MLP network structures.

4.2.2.2 Long Short Term Memory Networks

In addition to the three MLP networks, eight LSTM networks with different sets of hyperparameters (compare Table 3.2) were defined and trained. The hyperparameter setting of all trained LSTM networks, as well their respective loss values, are presented in Table 4.6. The output graphs for all LSTM networks are presented

in Appendices B.3 through B.10. Furthermore, the individual loss metrics for each output variable of all LSTM networks are presented in Appendix B, Table B.1.

	Number of layers	Neurons per layer	Data points per time step	Learning rate	Loss metric	Loss value
LSTM 1	2	50	3	0.001	MSE	0.002557
LSTM 2	2	100	3	0.001	MSE	0.002181
LSTM 3	2	150	3	0.001	MSE	0.002229
LSTM 4	2	250	3	0.001	MSE	0.002084
LSTM 5	2	500	3	0.001	MSE	0.002315
LSTM 6	4	50	3	0.001	MSE	0.003640
LSTM 7	6	150	50	0.0001	MSE	0.001335
LSTM 8	8	150	100	0.0001	MSE	0.001523

Table 4.6: Hyperparameter settings and loss values for each of the LSTM networks trained.

The first observation that can be made when comparing Table 4.4 with Table 4.6, is that almost all eight LSTM networks performed substantially better than the three MLP networks, with the poorest LSTM network (LSTM 6) performing only slightly worse than the best MLP network (MLP 2). A reasonable explanation for this difference in performance of LSTM and MLP networks is the occurrence of some time-dependant phenomena, such as fouling, which is intrinsically hard to capture by means of MLP networks. Consequently, this observation would be inline with the research question which was to identify if indication for time-dependent phenomena in the cooling tower system can be found. This becomes even more evident when comparing the best MLP network with the best LSTM network. In Figure 4.6 through Figure 4.9, the predicted model outputs of the best MLP network (MLP 2) and the best LSTM network (LSTM 7) are compared to the true output data (measurement data). The data is presented as a moving average over 200 data points.

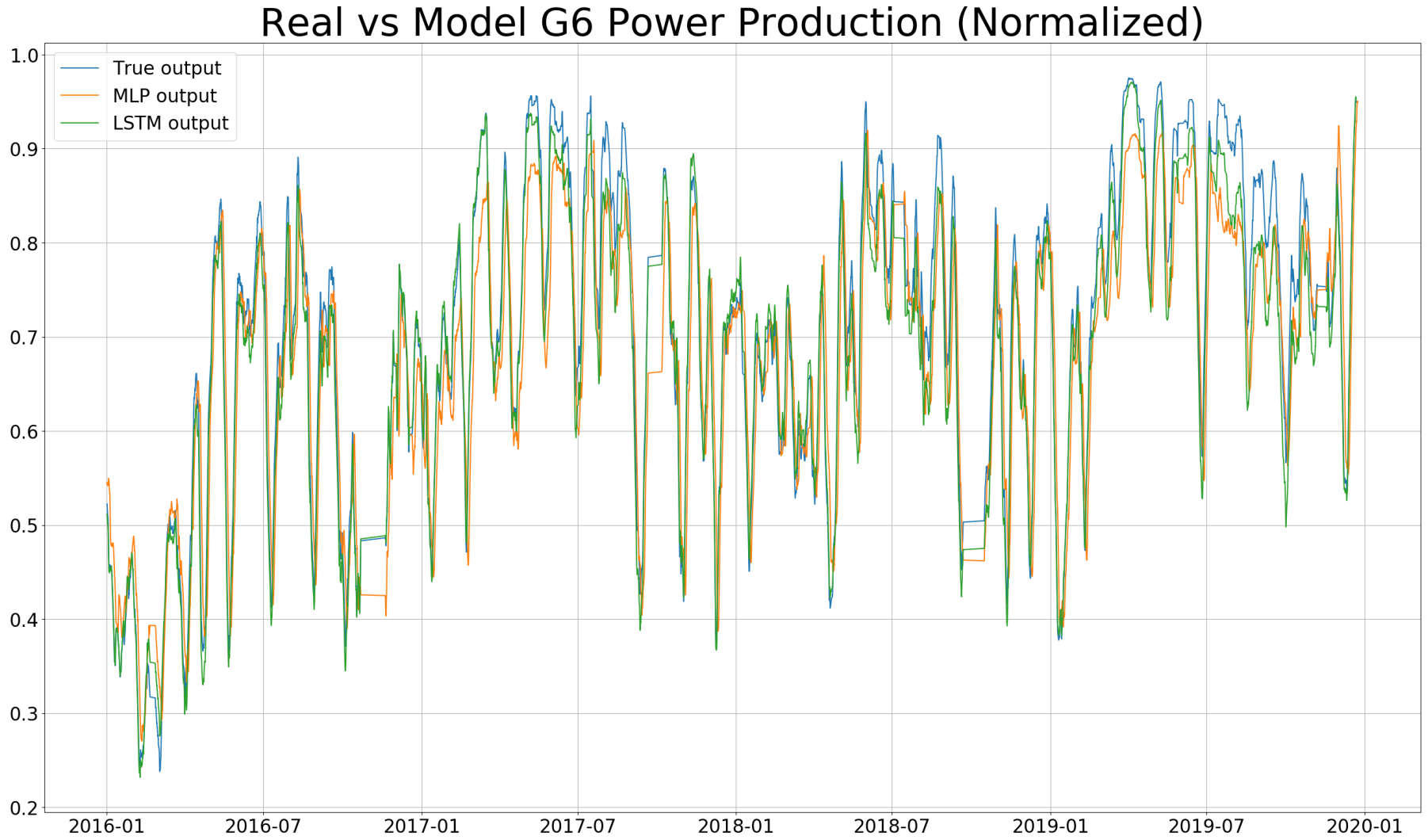

Figure 4.6: Comparison of power output between the best MLP network and the best LSTM network, as well as the true power output data.

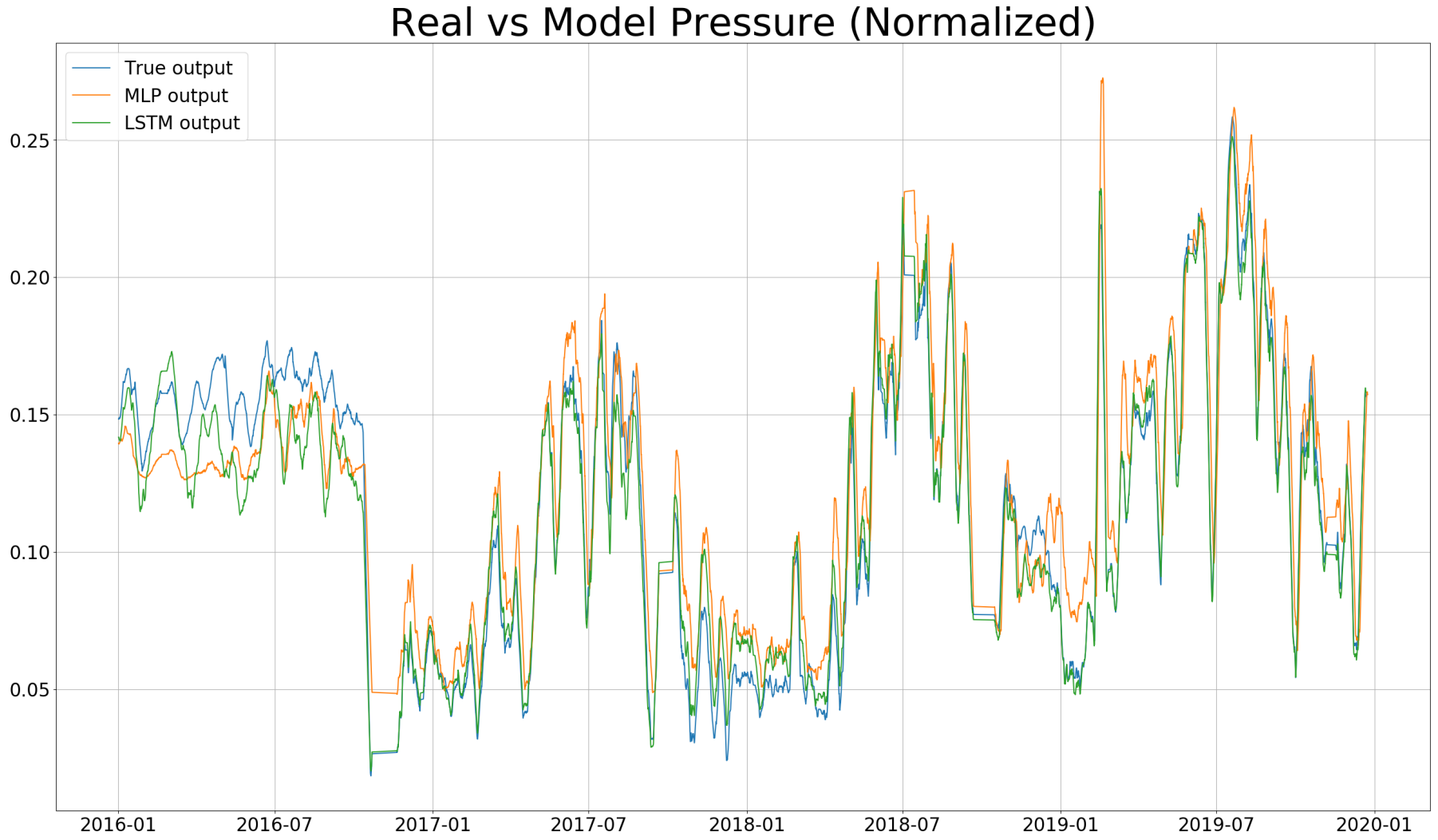

Figure 4.7: Comparison of pressure output between the best MLP network and the best LSTM network, as well as the true pressure output data.

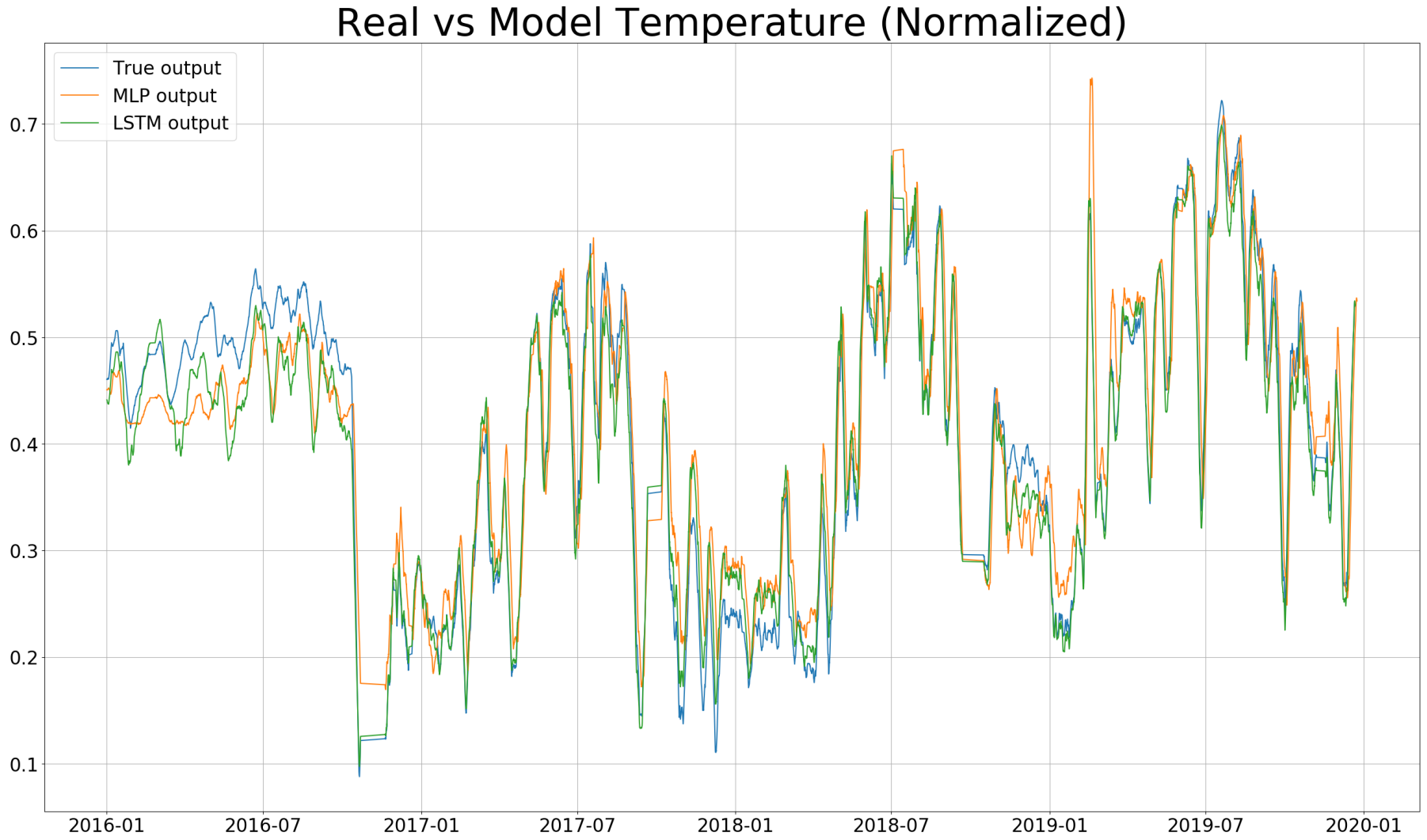

Figure 4.8: Comparison of temperature output between the best MLP network and the best LSTM network, as well as the true temperature output data.

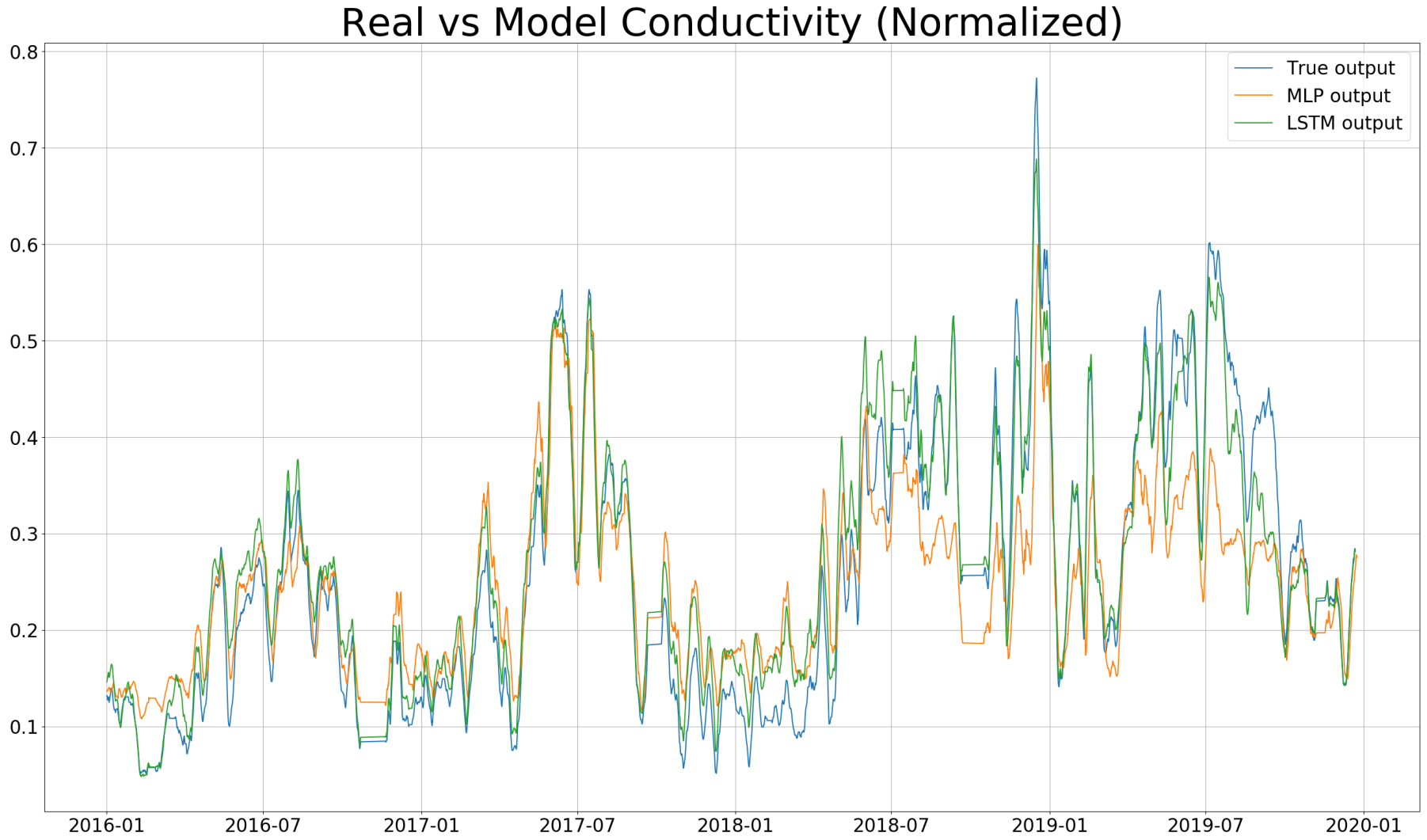

Figure 4.9: Comparison of conductivity output between the best MLP network and the best LSTM network, as well as the true conductivity output data.

In all Figures 4.6 through 4.9, the network LSTM 7 performs better than the network MLP 2, meaning that the LSTM network predicts all output variables with a greater accuracy than the MLP network. This is also observed by looking in Table 4.7, which presents the individual loss metric MSE for each of the output variables of MLP 2 and LSTM 7.

	Power production	**Pressure**	**Temperature**	**Conductivity**
MLP 2	0.002063	0.0004330	0.001963	0.008826
LSTM 7	0.001305	0.0002248	0.001313	0.002363

Table 4.7: Individual MSE loss metric values for each output variable of networks MLP 2 and LSTM 7.

For the output variables power production, pressure and temperature, rather small discrepancies between the MLP network and the LSTM network can be observed in Figures 4.6 through 4.8. This is also indicated by Table 4.7, as the loss metrics of these three output variables (power production, pressure and temperature) are around 30 % to 50 % lower for the LSTM network than for the MLP network. Figure 4.7 and Figure 4.8 further indicate that the MLP network overshoots the peaks in the data, observed around 2019-01 on the x-axis. The most obvious difference in accuracy between the MLP network and the LSTM network can be observed in Figure 4.9, where the conductivity output is compared. In the beginning, indicated by the left side of the graph, the difference between the two networks is rather small, with the LSTM network performing slightly better. However, as time passes, indicated by the right side of the graph, the discrepancy between the conductivity outputs of LSTM 7 and MLP 2 is increasing, with there being an enormous difference in accuracy towards the end. This discrepancy is also confirmed in Table 4.7 as the loss metric for conductivity is almost four times larger for MLP 2 compared to LSTM 7.

4.2.3 Validation of LSTM 7

The best performing model, LSTM 7, was further validated by using data from 2020-01-01 to 2020-05-15. The model was used to predict the values of the output variables which then was compared to the extracted data from the pulp mill's monitoring system. The data is presented as a moving average of 50 data points, where the blue graph illustrates the extracted data and the green line illustrates the output of LSTM 7.

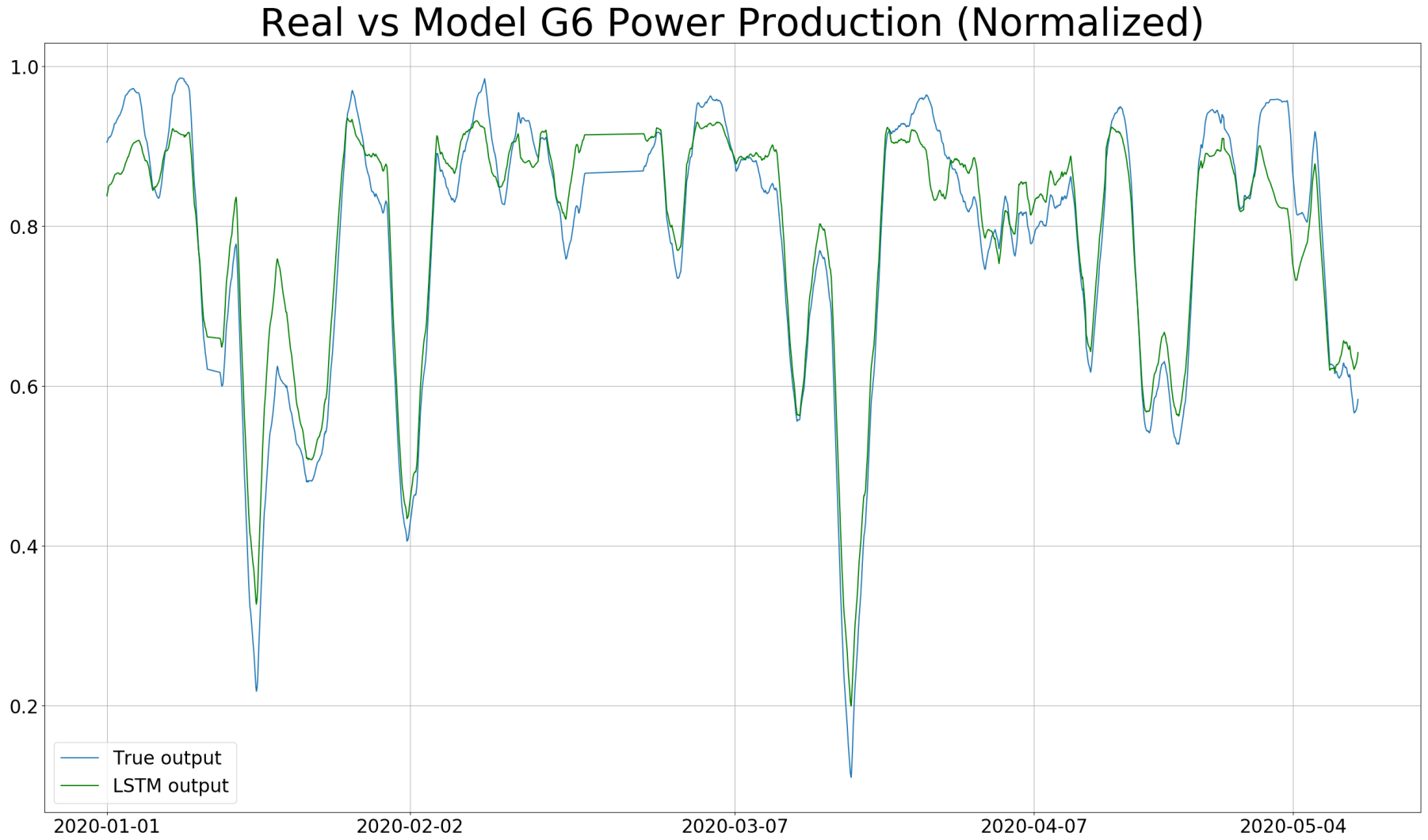

Figure 4.10: Power production output of LSTM 7, compared to the true output. Validation on data extracted from beginning of January 2020 to middle of May 2020.

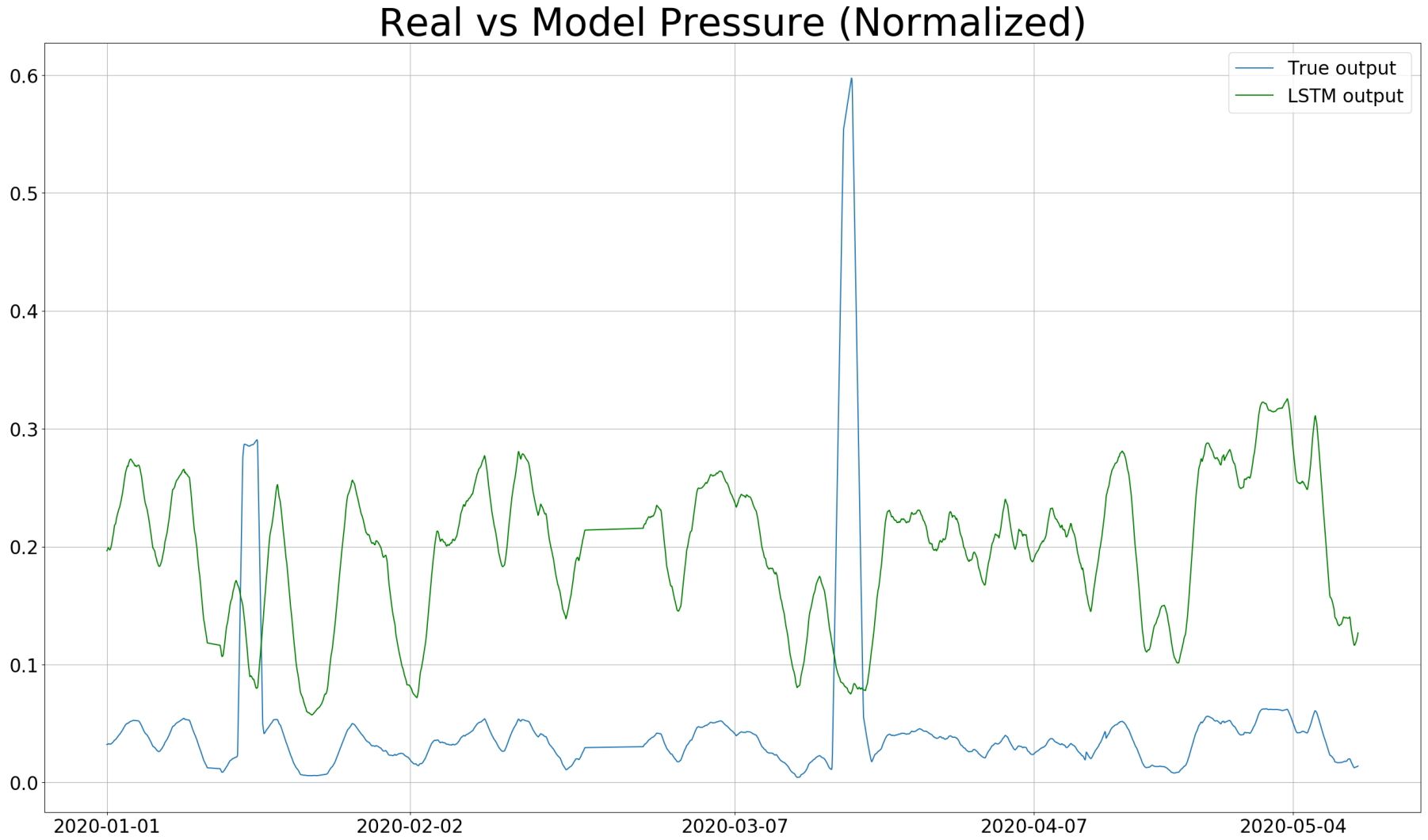

Figure 4.11: Pressure output of LSTM 7, compared to the true output. Validation on data extracted from beginning of January 2020 to middle of May 2020.

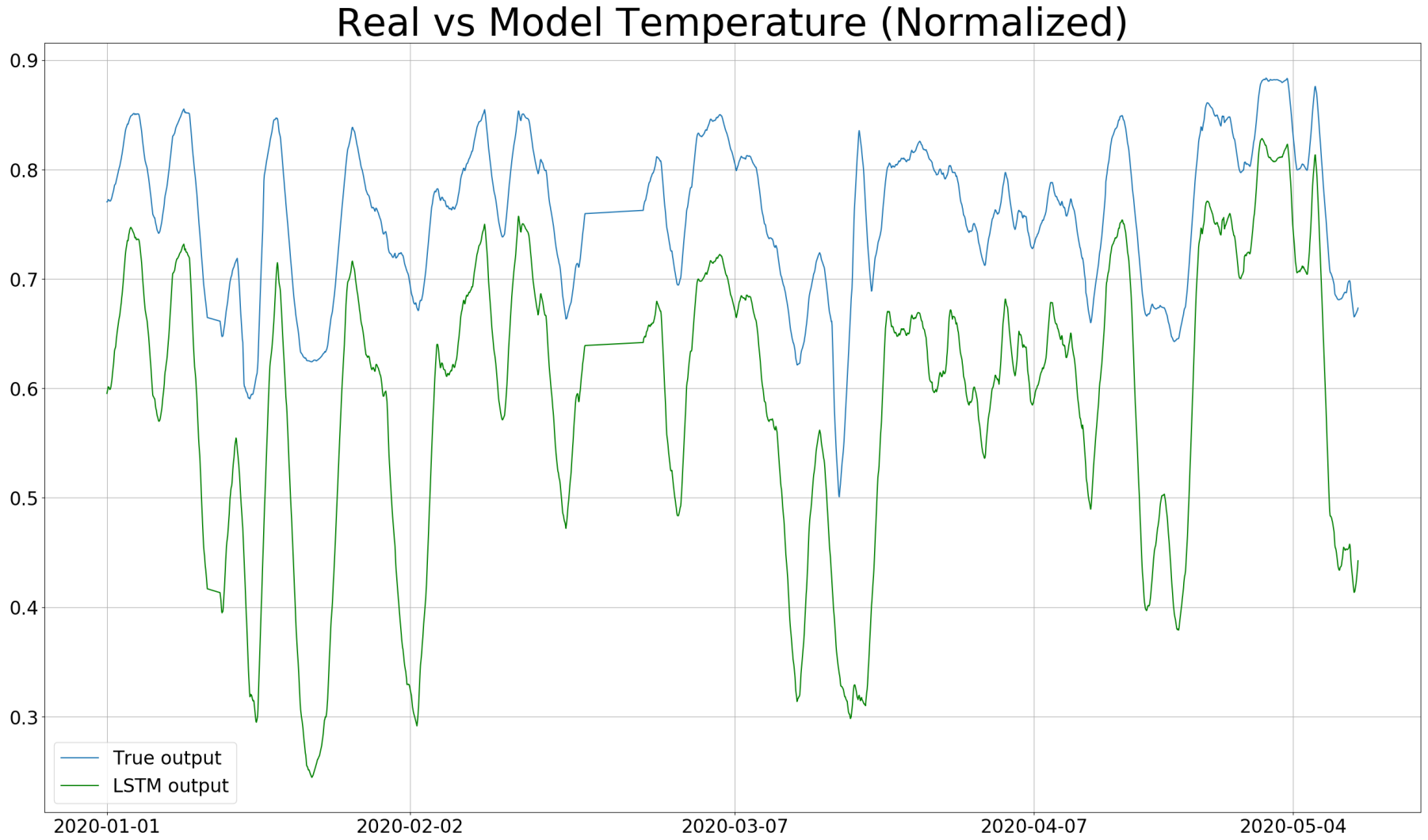

Figure 4.12: Temperature output of LSTM 7, compared to the true output. Validation on data extracted from beginning of January 2020 to middle of May 2020.

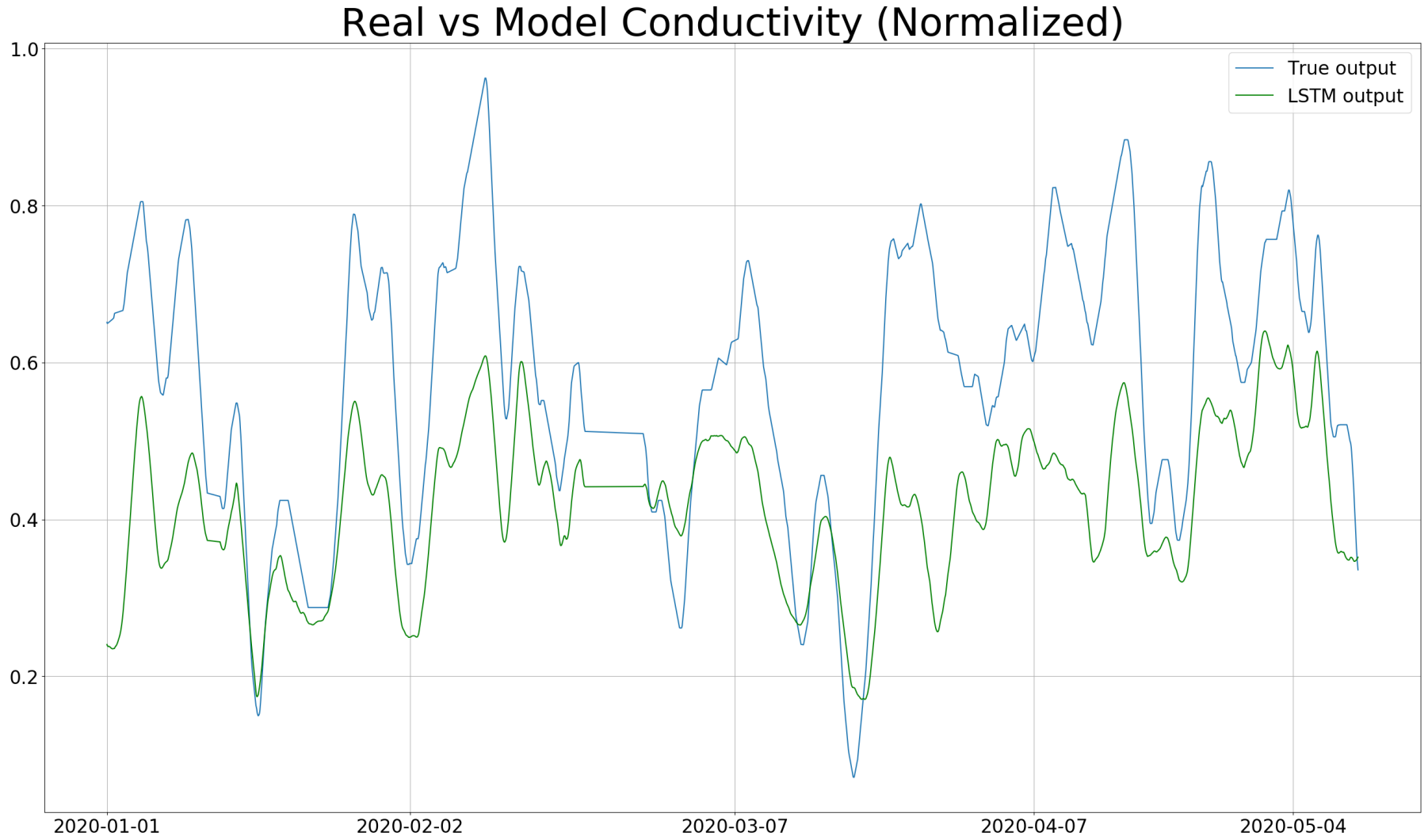

Figure 4.13: Conductivity output of LSTM 7, compared to the true output. Validation on data extracted from beginning of January 2020 to middle of May 2020.

The validation process indicates that LSTM 7 can successfully predict power production, as observed in Figure 4.11. In contrast, the three remaining output variables, i.e. pressure, temperature and conductivity, are predicted poorly, as observed in Figures 4.11 through 4.13. This becomes more evident when analysing Table 4.8, which presents the loss metric values for each output variable of the model.

	Power production	**Pressure**	**Temperature**	**Conductivity**
LSTM 7	0.001188	0.02687	0.07297	0.07096

Table 4.8: Individual MSE loss metric values for each output variable of LSTM 7.

The only loss value which is comparable, in terms of accuracy, with the values presented in Table B.1 and Table 4.5, is the power production. Interestingly, the loss value of power production obtained during the validation process, is the lower than the loss value obtained in training. For the remaining output variables, their loss values are one or two magnitudes of order larger than the loss values in training.

5
Discussion

This chapter will present the reader with discussions regarding the various topics treated in this project. First, a discussion regarding the quality and availability of data at Södra Cell Mönsterås is presented, followed by some points and thoughts that occurred during the training and evaluation of machine learning techniques. This chapter also presents the reader with some insights into working with machine learning, and what could have been done to improve the outcome of the project work.

5.1 Södra Cell Mönsterås and Data

During the meetings with plant operators and engineers at Södra Cell Mönsterås, one topic that was brought up was the quality of data available. When asked, the plant operators mentioned that the accuracy of their current sensors and measurement devices was far from perfect. Especially flow measurement devices were known for their poor accuracy, which is why there were some sensors using calculated values instead. These calculations utilised the measured values together with mass balances, which resulted in more accurate values. Wherever available in the system, these calculated values were used.

One other topic that was discussed was the time period of extracted data. As mentioned, data was extracted from a time period between the beginning of 2016 and the beginning of 2020, with a time resolution of one hour. During these four years, several events occurred at the pulp mill. In this context, events refer to installation of new equipment, changes in maintenance routines and so forth. It would have been interesting to get information regarding the events occurring during these four years, as it might have been possible to assign the cause of certain trends in data, to these events. However, this information could not be provided.

5.2 Training and evaluation of MLP and LSTM

Table 4.4 and Table 4.6 both show that there were clear differences in model accuracy of the two machine learning techniques, as well as there being a differences when hyperparameters were varied. In Appendix B, the output graphs for all LSTM models are presented. While Table 4.6 indicates that there is a difference in accuracy between the models, it can be hard to determine from the graphs. Therefore, one should observe Table B.1. According to the table, LSTM 7 seems to more accurately

predict the conductivity, while LSTM 8 models the power output to a better extent, even though LSTM 7 obtained better model accuracy. One general trend that can be observed in Table B.1 is that LSTM 1 through LSTM 5 managed to predict power production with much greater accuracy than networks LSTM 6 through LSTM 8. However, LSTM 7 and LSTM 8 predicted conductivity better than the other LSTM networks. LSTM 1 through LSTM 5 all consisted of two hidden layers with three data points per time step, with varying amounts of neurons per layer, while LSTM 6 through LSTM 8 consisted of varying amounts of layers and data points per time step. This indicates that the variation of hyperparameters has a distinct effect on the outcome of the system model. Adding neurons to the layers seem to lead to greater power production accuracy, which can be observed by comparing networks LSTM 1 through LSTM 4 and comparison of LSTM 7 with LSTM 8, in Table B.1. On the other hand, improvements in conductivity prediction is mainly seen through added model complexity, i.e. adding layers to the network and increasing the amount of data points per time step. However, this trend of improved model accuracy is only observed to a certain extent, as LSTM 8 predicts conductivity worse than LSTM 7, even though LSTM 8 is more complex as the model consists of more hidden layers than LSTM 7 as well as more data points per time step.

Furthermore, the plant operators explained that there exist several ways of controlling the power production. One such way was by controlling the pressure and temperature at the outlet. By forcing the steam to exit the turbine at a higher pressure and temperature, it also leaves at a higher enthalpy. This means that the difference in enthalpy is reduced, resulting in less power production, as the production of power is defined by:

$$Q = \dot{m}_{steam} \times (h_{steam,in} - h_{steam,out})$$

Naturally, this means that pressure and temperature follow the same trend line over time, which can be observed in all output graphs. In hindsight, it might have been better, in terms of model accuracy, to remove either pressure or temperature as an output variable. Additionally, power production is, as explained, a consequence of the flow of steam through the turbine, the enthalpy at the inlet, and the enthalpy at the outlet. The enthalpy and steam flow at the inlet was accounted for by the input variables, and the enthalpy at the outlet was accounted for by the output variables. This means that, technically, power production could be removed as an outlet with the purpose of improving the model accuracy for the remaining output variables.

One of the most interesting hyperparameters to study was the learning rate. As described in Chapter 2, the learning rate determines the magnitude of changes to the weight factors in the network. Therefore, it plays a crucial role in whether or not the algorithm manages to properly minimise the loss function, especially when model complexity is increased. For example, there might be a neural network with two layers with 50 neurons each. This results in a given amount of connections, and thus weight factors, which model the dataset of interest. If the complexity is increased, either by adding more neurons to each layer, adding more layers, or both, there are now substantially more weight factors that are modeling the same dataset.

Naturally, this means that the weight factors cannot remain at similar values, as each weight factor has a lower impact on the network overall. Therefore, if model complexity is increased to a certain extent, without lowering the learning rate, one can obtain models with unstable behaviour that eventually diverge completely.

Technically, one could program a complex network with a very small value for the learning rate, and give it enough training time to properly minimise the loss function. There is one problem with this approach; computational capacity, or training time. Out of curiosity, one such complex model was trained on a powerful personal computer. The network had the following hyperparameter settings:

	Number of layers	Neurons per layer	Data points per time step	Learning rate	Loss metric
LSTM Test network	8	250	500	0.00001	MSE

Table 5.1: Hyperparameter settings for a LSTM test network.

For this network, each training cycle, called epoch, took over three hours. Due to the small learning rate, more training cycles are needed, and it was estimated that training required around 300 to 400 epochs for the loss function to be properly minimised. This means around 1000 hours, or more than 40 days, of model training. While a more powerful computer can do this training in a shorter time, it becomes obvious that it is not feasible to train networks for such a long time. One also has to be reminded of the main purpose of the modeling approach. In this project, the main purpose was to investigate whether or not there was a time-dependant phenomena present in the system. Obviously, model accuracy was of importance, but only to the extent that the model could fulfil its purpose.

As for the validation process, it was previously mentioned that power production is a consequence of several variables, including temperature and pressure of the steam at the turbine outlet. Therefore, it was quite interesting that power production was predicted with good accuracy meanwhile the remaining variables were predicted quite poorly. Generally, power production is the variable of greatest interest to predict, and if this can be done even though the loss values of the remaining output variables, and also overall loss value of the model, are poor, it shows the potential of improving such a model in future works. To summarise, in this thesis data-driven modelling was carried out, yielding an accurate statistical model of a cooling tower system at the Södra Cell Mönsterås pulp mill. This model, or an improved version of it, could be used further on in future works, where some kind of optimisation would be carried out. Examples of such future projects are presented in Chapter 7.

6
Conclusion

The main aim of this project was to investigate and approach process industry challenges from a data-driven perspective, which in this project referred to the application of various machine learning methods. Reviewing the literature showed that there is definitely usefulness in approaching process industry issues from a data-driven perspective. The literature review identified several studies in which various problems in industry were investigated via the means of machine learning techniques, with successful outcomes.

The results from the case study of a pulp mill cooling tower system investigated in this project also showed the benefits of approaching problems from a data-driven perspective. The machine-learning approach allowed for efficient data-driven modelling of the cooling tower system without the need for explicit equation-based process models, and the resulting models described the system with high accuracy.

The results of the case study showed that there is some time-dependant phenomena, such as fouling, affecting the cooling tower system. This confirmed an hypothesis of potential fouling effects, which was suggested by plant operators at Södra Cell Mönsterås. This can be argued for in several ways. First of all, the results indicate that there is a clear difference in model accuracy between two investigated architectures of neural networks; MLP networks and LSTM networks. This means that there is some phenomena of recurrency present in the dataset, which the MLP networks can not account for, but which is captured by the LSTM networks.

Each set of data points represents the variables in the dataset at a given time, as the data was extracted with respect to time, with the time resolution of one hour. The recurrent behaviour indicates that the outcome of a given set of data points influence the next set of data points to a certain degree. For the dataset used in this master thesis project, it means that the outcome of a given point in time influences the outcome of a point in time in the future. The MLP networks do not consider recurrency in any way, meaning that such a network only tries to fit a set of input variables to a set of output variables, with no respect to the sequence of time. The LSTM networks however, do consider recurrency by processing the data in a time-sequential order, and allowing for the previous outcome to influence the following outcomes. Thus, the fact that the LSTM networks performed better overall lead to the conclusion that there is some time-dependant phenomena present in the system.

Resulting model output trends confirms the occurrence of some time-dependant phenomena. The discrepancy between the LSTM model and the MLP model grows, as time passes, with the LSTM model always performing better than the MLP model. The passage of time means that more of the time-dependant phenomena, or recurrency, is allowed to influence the system. The MLP network cannot account for this influence, meanwhile the LSTM network can, resulting in greater differences in model accuracy between the two architectures, as time passes.

The overall findings of this project work lead to the conclusion that there are various process industry challenges that can be approached with the means of machine learning, successfully. Due to the powerful data processing properties of machine learning techniques, they allow for completely new findings that could previously not been observed or confirmed. The confirmation of time-dependent fouling effects in the pulp mill cooling tower system is an example of such new findings.

7
Future work

During the course of this project work, several ideas arose regarding possible future projects. In this project, the time-dependant phenomena occurring in the cooling tower system was successfully modeled. These results could possibly be used as groundwork for future work. One example of such a project would be the optimisation of maintenance interventions to improve operational conditions. Based on data from various scenarios, such as typical weather conditions and varying forestry yields, the power production, which is based on the cooling tower performance, could be predicted. Based on the model for predicted values of power production, an optimisation of maintenance decisions could be performed with regard to overall cost efficiency. Such an optimisation problem could be solved to schedule maintenance of the cooling tower in order to improve cooling performance. Subsequently, this would lead to greater production of power which means extra electricity that can be sold to the surrounding grid.

Other possible future projects also relate to the scheduling of maintenance, but for other unit processes in the mill. One case study, which was discussed with operators and engineers at Södra Cell Mönsterås, but not further analysed, was the evaporator system. The operators at the mill have had two different cleaning routines for the system, but never evaluated which one was superior. By processing data during the time periods in which these two cleaning routines were implemented, and observing trends for key performance indicators such as overall heat transfer coefficients, the optimal cleaning routine could be determined.

Bibliography

[1] UNFCCC, *What is the Paris Agreement? | UNFCCC*. [Online]. Available: https://unfccc.int/process-and-meetings/the-paris-agreement/what-is-the-paris-agreement (visited on 28/01/2020).

[2] European Parliament, "DIRECTIVE (EU) 2018/2002 amending Directive 2012/27/EU on Energy Efficiency", *Official Journal of the European Union*, vol. 328, no. November, pp. 210–230, 2018. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32018L2002%7B%5C&%7Dfrom=EN.

[3] European Comission, "The revised energy efficiency directive", p. 2030, 2018. [Online]. Available: https://ec.europa.eu/energy/sites/ener/files/documents/energy%7B%5C_%7Defficiency%7B%5C_%7Dfactsheet.pdf.

[4] Energimyndigheten, "Sektorsstrategier för energieffektivisering Sverige ska bli världsbäst på", Statens energimyndighet, Stockholm, Tech. Rep., 2018, p. 107.

[5] ——, *Energitillförsel och energianvändning i Sverige 2018, TWh*, 2018. [Online]. Available: https://www.energimyndigheten.se/globalassets/statistik/officiell-statistik/statistikprodukter/arlig-energibalans/ovrigt/databars2018%7B%5C_%7Dsv.pdf (visited on 28/01/2020).

[6] ——, "Energiläget 2017", p. 86, 2017. [Online]. Available: www.energimyndigheten.se.

[7] Södra Cell Mönsterås, *Södra Cell Mönsterås*. [Online]. Available: https://www.sodra.com/sv/massa/vara-massabruk/monsteras/fakta-om-sodra-cell-monsteras/ (visited on 29/01/2020).

[8] ——, *Effektivt resursnyttjande*, 2018. [Online]. Available: https://www.sodra.com/sv/hallbarhet/strategi-for-hallbarhet/effektivt-resursnyttjande/ (visited on 29/01/2020).

[9] F. Nihlmark and M. Mahmoud, "Analysis of the Secondary Heating System of Södra Cell Mönsterås", Chalmers University of Technology, Gothenburg, Tech. Rep., 2017.

[10] P. Bokinge and D. Erlandsson, "Evaluation of the flexibility of large and complex heat exchanger networks", Chalmers University of Technology, Gothenburg, Tech. Rep., 2018.

[11] J. Persson and T. Berntsson, "Influence of seasonal variations on energy-saving opportunities in a pulp mill", *Energy*, vol. 34, no. 10, pp. 1705–1714, 2009, ISSN: 03605442. DOI: 10.1016/j.energy.2009.07.023. [Online]. Available: http://dx.doi.org/10.1016/j.energy.2009.07.023.

[12] ——, "Influence of short-term variations on energy-saving opportunities in a pulp mill", *Journal of Cleaner Production*, vol. 18, no. 9, pp. 935–943, 2010,

ISSN: 09596526. DOI: 10.1016/j.jclepro.2009.12.018. [Online]. Available: http://dx.doi.org/10.1016/j.jclepro.2009.12.018.

[13] Merriam-Webster, *Machine Learning | Definition of Machine Learning by Merriam-Webster*. [Online]. Available: https://www.merriam-webster.com/dictionary/machine%20learning (visited on 29/01/2020).

[14] A. Ragab, M. El-Koujok, B. Poulin, M. Amazouz and S. Yacout, "Fault diagnosis in industrial chemical processes using interpretable patterns based on Logical Analysis of Data", *Expert Systems with Applications*, vol. 95, pp. 368–383, 2018, ISSN: 09574174. DOI: 10.1016/j.eswa.2017.11.045. [Online]. Available: https://doi.org/10.1016/j.eswa.2017.11.045.

[15] J. J. Costello, G. M. West and S. D. McArthur, "Machine learning model for event-based prognostics in gas circulator condition monitoring", *IEEE Transactions on Reliability*, vol. 66, no. 4, pp. 1048–1057, 2017, ISSN: 00189529. DOI: 10.1109/TR.2017.2727489.

[16] M. Sainlez and G. Heyen, "Comparison of supervised learning techniques for atmospheric pollutant monitoring in a Kraft pulp mill", *Journal of Computational and Applied Mathematics*, vol. 246, pp. 329–334, 2013, ISSN: 03770427. DOI: 10.1016/j.cam.2012.06.026.

[17] D. Plattformar, "Swedish IndTech",

[18] *Recurrent Neural Networks - Towards Data Science*. [Online]. Available: https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce (visited on 17/05/2020).

A

Appendix 1

A.1 Filtering

```
RemoveIndex = labels.query('MW<5').index
IndexRemove = [list(map(itemgetter(1), g)) for k, g in\
               groupby(enumerate(RemoveIndex), \
                       lambda x: x[0]-x[1])]
IndexRemove1 = len(IndexRemove)

for arbitrary in range(IndexRemove1):
    if IndexRemove == []:
        break

    if IndexRemove[0][0]<10:
        data = data.drop(data.index\
               [IndexRemove[0][0]:IndexRemove[0][-1]+10])
        labels = labels.drop(labels.index\
                 [IndexRemove[0][0]:IndexRemove[0][-1]+10])

    if IndexRemove[0][-1]>len(data)-10:
        data = data.drop(data.index[IndexRemove\
                         [0][0]-10:IndexRemove[0][-1]])
        labels = labels.drop(labels.index[IndexRemove\
                             [0][0]-10:IndexRemove[0][-1]])

    data = data.drop(data.index[IndexRemove[0][0]-10:\
                                IndexRemove[0][-1]+10])
    data = data.reset_index()
    data = data.drop(['index'], axis=1)

    labels = labels.drop(labels.index[IndexRemove[0][0]-10\
                                :IndexRemove[0][-1]+10])
    labels = labels.reset_index()
    labels = labels.drop(['index'], axis=1)

    RemoveIndex = labels.query('MW<5').index
    IndexRemove = [list(map(itemgetter(1), g)) for k, g in\
```

```
                groupby(enumerate(RemoveIndex), \
                    lambda x: x[0]-x[1])]
```

A.2 Normalisation

```
transformer = MinMaxScaler()
transformer.fit(dataset)
dataset = transformer.transform(dataset)
```

A.3 MLP networks

The MLP networks were constructed by the following code,

```
model = Sequential()
model.add(Dense(500, activation='relu',
                input_dim=len(data[0,:])))
model.add(Dense(500, activation='relu'))
model.add(Dense(500, activation='relu'))
model.add(Dense(500, activation='relu'))
model.add(Dense(len(labels[0,:])))
model.compile(optimizer='adam', loss='mse')
```

where each line of

```
model.add(Dense(500, activation='relu'))
```

represents one hidden layer of the neural network, and the number 500 indicates the amount of neurons in that layer. In the code presented, this would mean a MLP network with four layers and 500 neurons per layer. Finally, the MLP network was trained through the following line of code:

```
model.fit(data, labels, epochs=200, verbose=1)
```

A.4 LSTM networks

For the LSTM networks, one additional pre-processing step had to be done. As described in section 2.2.1.3, the LSTM networks process data in batches, or time steps. Thus, the data had to be divided into batches prior to training, which was done by,

```
def split_sequences(sequences, n_steps):
    X, y = list(), list()
    for i in range(len(sequences)):

        end_ix = i + n_steps

        if end_ix > len(sequences):
            break
        seq_x, seq_y = sequences[i:end_ix, \
                                 :-len(labels.columns)],\
            sequences[end_ix-1, -len(labels.columns):]
        X.append(seq_x)
        y.append(seq_y)
    return np.array(X), np.array(y)

n_steps = 10

X, y = split_sequences(dataset, n_steps)
```

where the size of each batch, or time step, was defined by:

```
n_steps = 10
```

The LSTM network was then constructed with the following code:

```
model = Sequential()
model.add(LSTM(150, activation='relu', return_sequences=True,
               input_shape=(n_steps, n_features)))
model.add(LSTM(150, activation='relu',
               return_sequences=True))
model.add(LSTM(150, activation='relu',
               return_sequences=True))
model.add(LSTM(150, activation='relu',
               return_sequences=True))
model.add(LSTM(150, activation='relu',
               return_sequences=True))
model.add(LSTM(150, activation='relu'))
model.add(Dense(n_output))
model.compile(optimizer=optimizer, loss='mse')
```

This code would result in a LSTM network consisting of six layers with 150 neurons in each layer and a batch size of ten data points.

Similar to the MLP networks, the number of hidden layers could be changed by removing or adding the line

```
model.add(LSTM(150, activation='relu',
               return_sequences=True))
```

and the number of neurons per layer could be varied by changing the number 150.

For the LSTM networks, the learning rate was also studied. The learning rate was changed by applying the following line of code prior to defining the model architecture:

```
optimizer = tf.keras.optimizers.Adam(lr=0.0001)
```

B

Appendix 2

Here, the model output graphs of all MLP networks and LSTM networks are presented. In Table B.1, the individual MSE values for each output variable for all LSTM networks, are presented.

	Power production	**Pressure**	**Temperature**	**Conductivity**
LSTM 1	0.0005680	0.0004383	0.002469	0.006662
LSTM 2	0.0005319	0.0002569	0.001509	0.007138
LSTM 3	0.0004766	0.0002182	0.001478	0.007489
LSTM 4	0.0004255	0.0003338	0.001817	0.005995
LSTM 5	0.0006045	0.0002300	0.001243	0.008227
LSTM 6	0.002530	0.0005851	0.003069	0.008651
LSTM 7	0.001305	0.0002248	0.001313	0.002363
LSTM 8	0.0008296	0.0002067	0.001164	0.004253

Table B.1: Individual MSE loss metric values for each output variable of all LSTM networks.

B.1 MLP 1

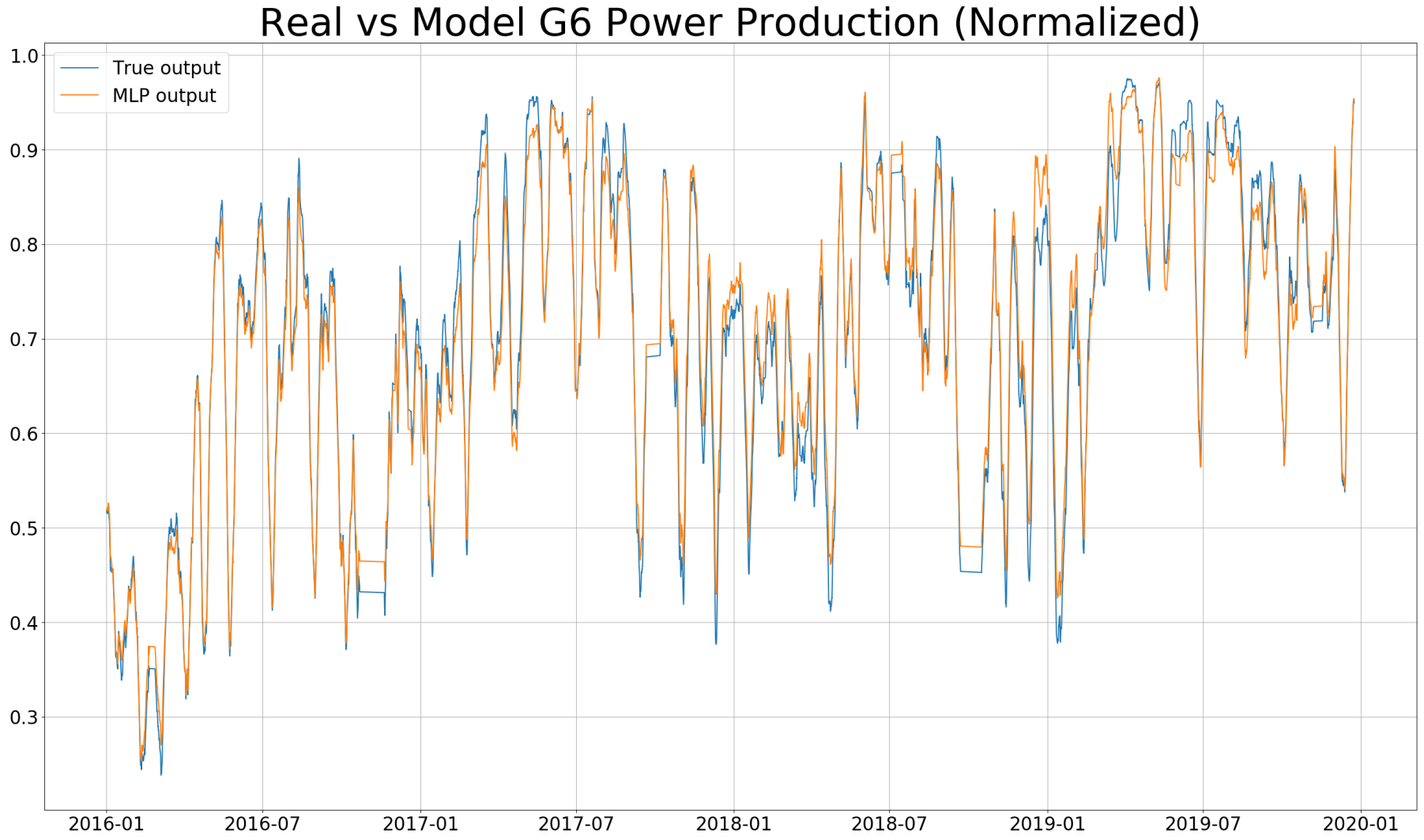

Figure B.1: Power production output of MLP 1, compared to the true output.

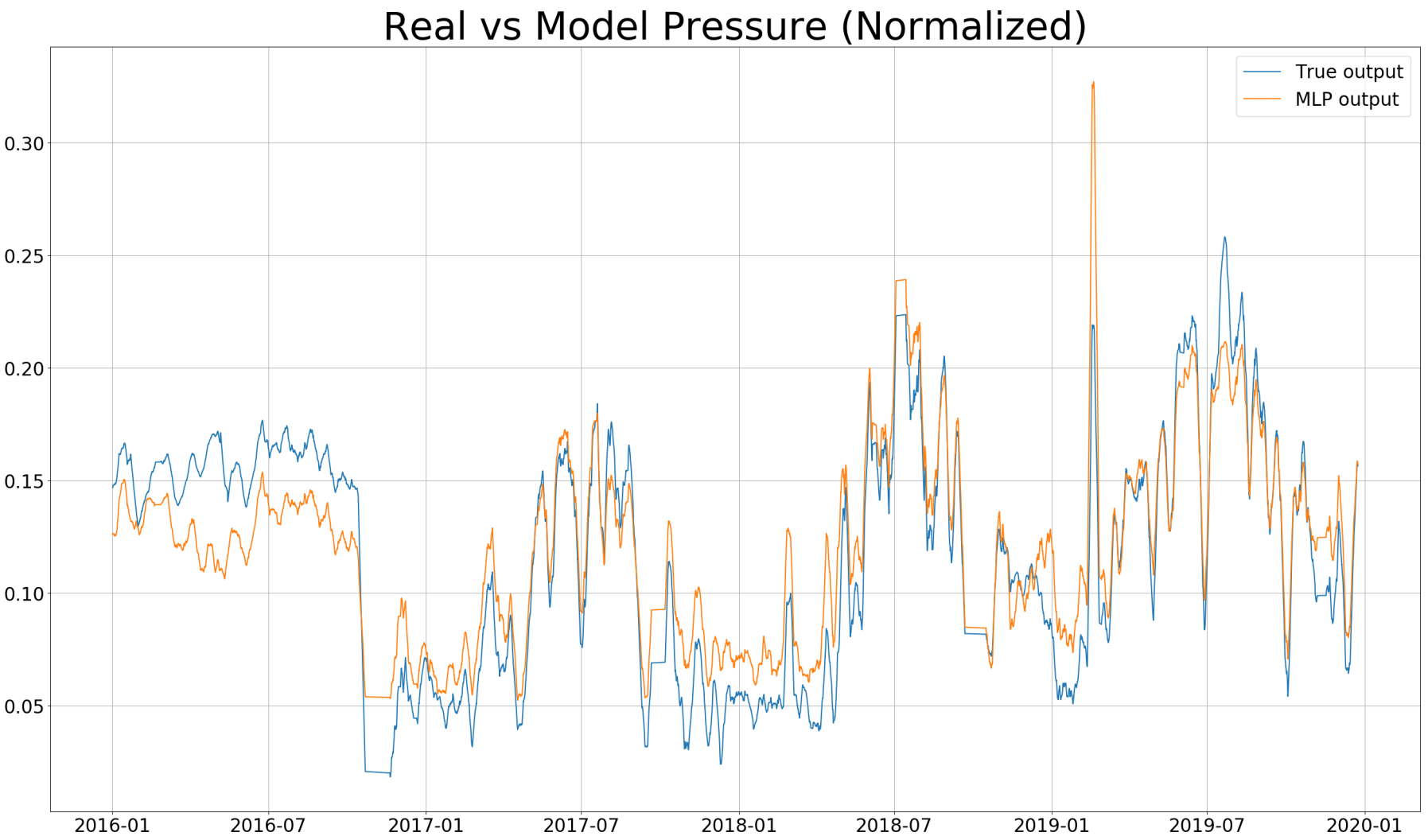

Figure B.2: Pressure output of MLP 1, compared to the true output.

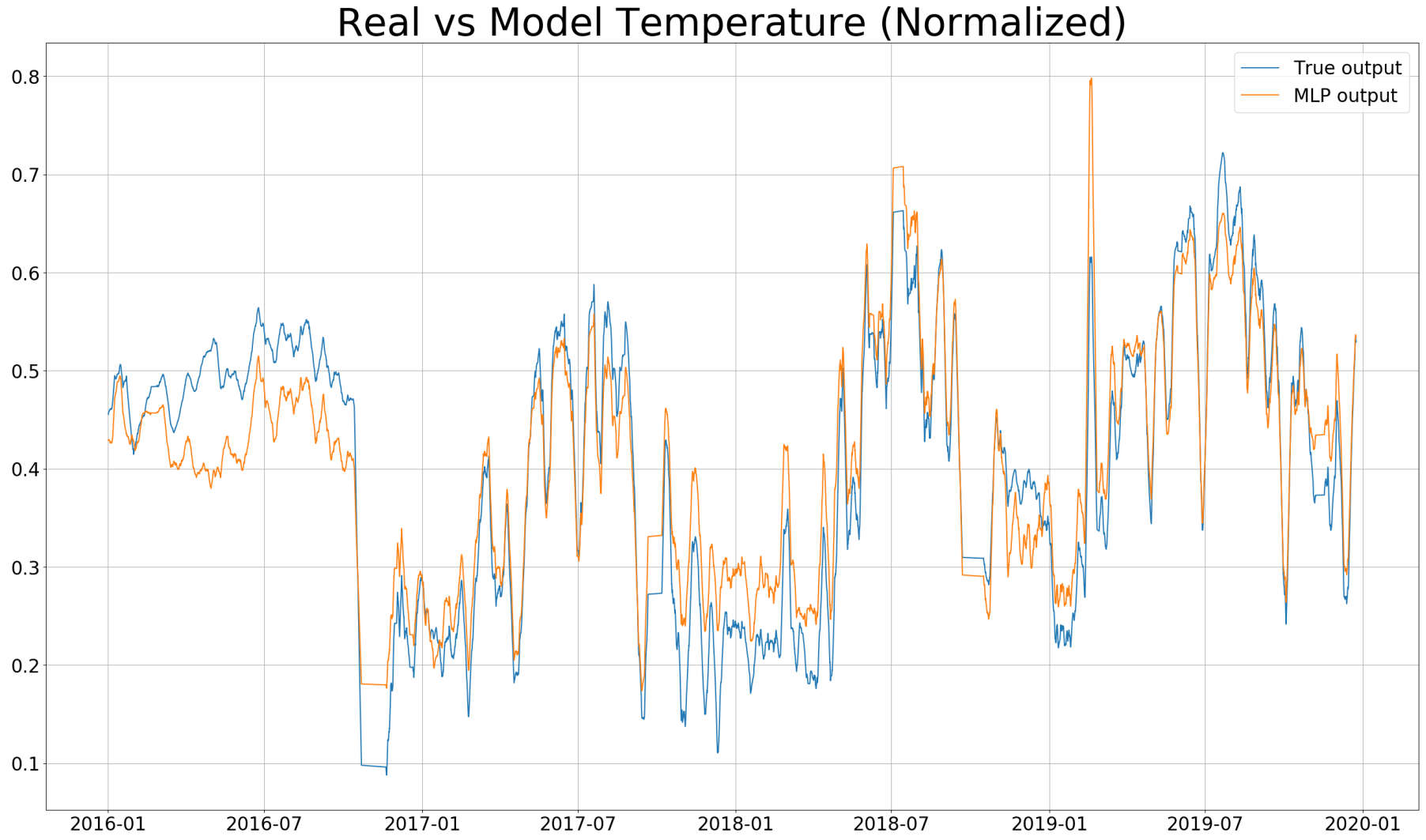

Figure B.3: Temperature output of MLP 1, compared to the true output.

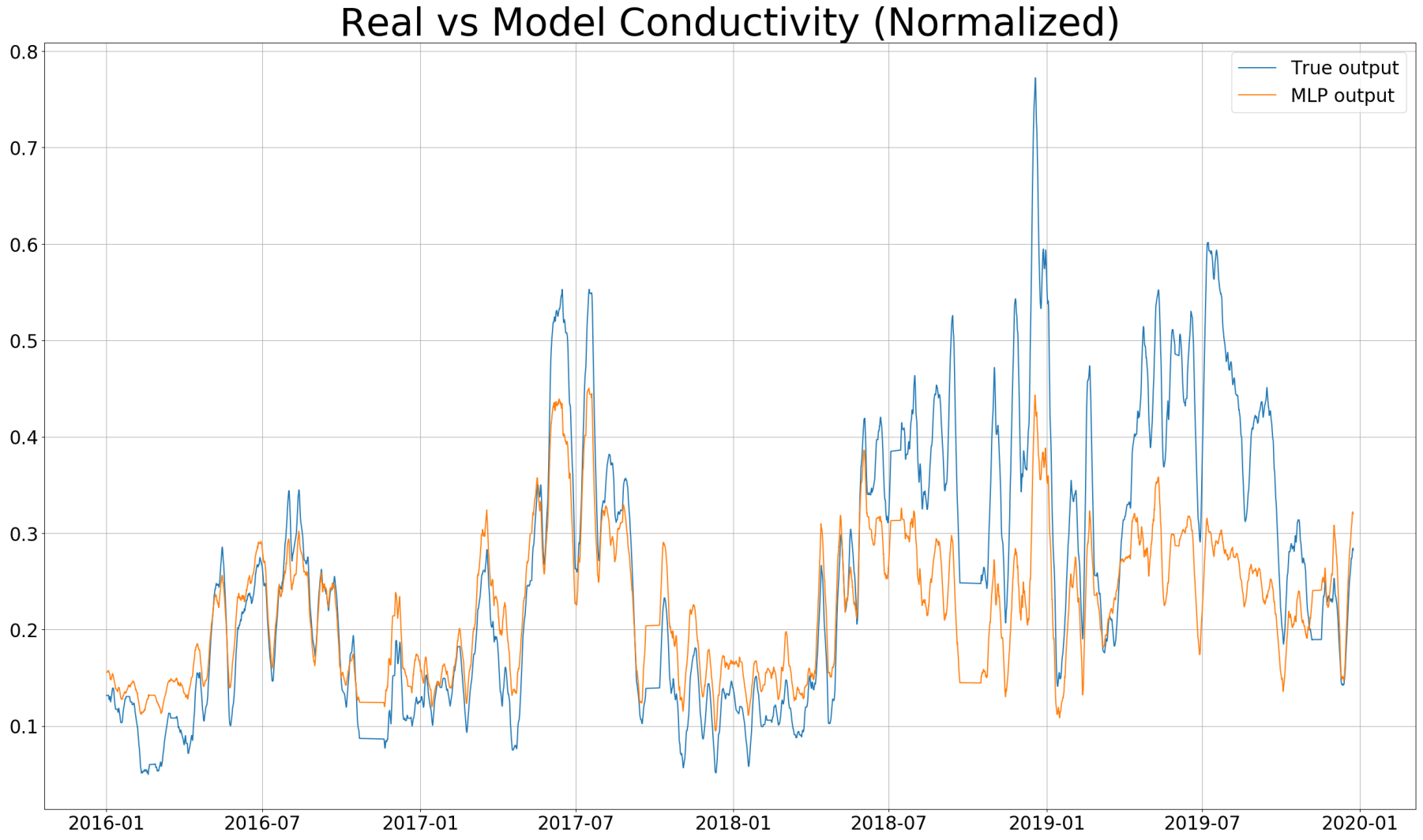

Figure B.4: Conductivity output of MLP 1, compared to the true output.

B.2 MLP 3

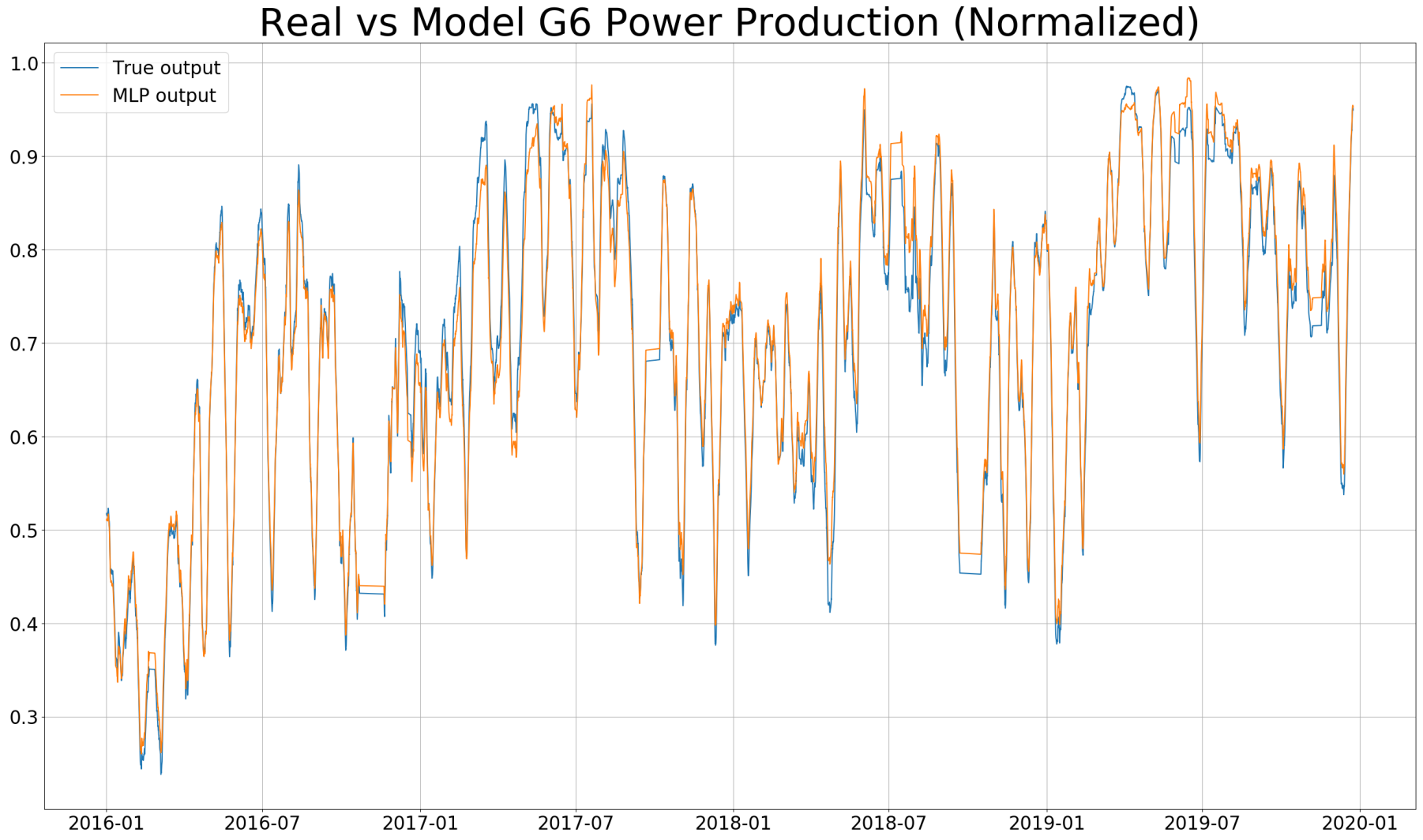

Figure B.5: Power production output of MLP 3, compared to the true output.

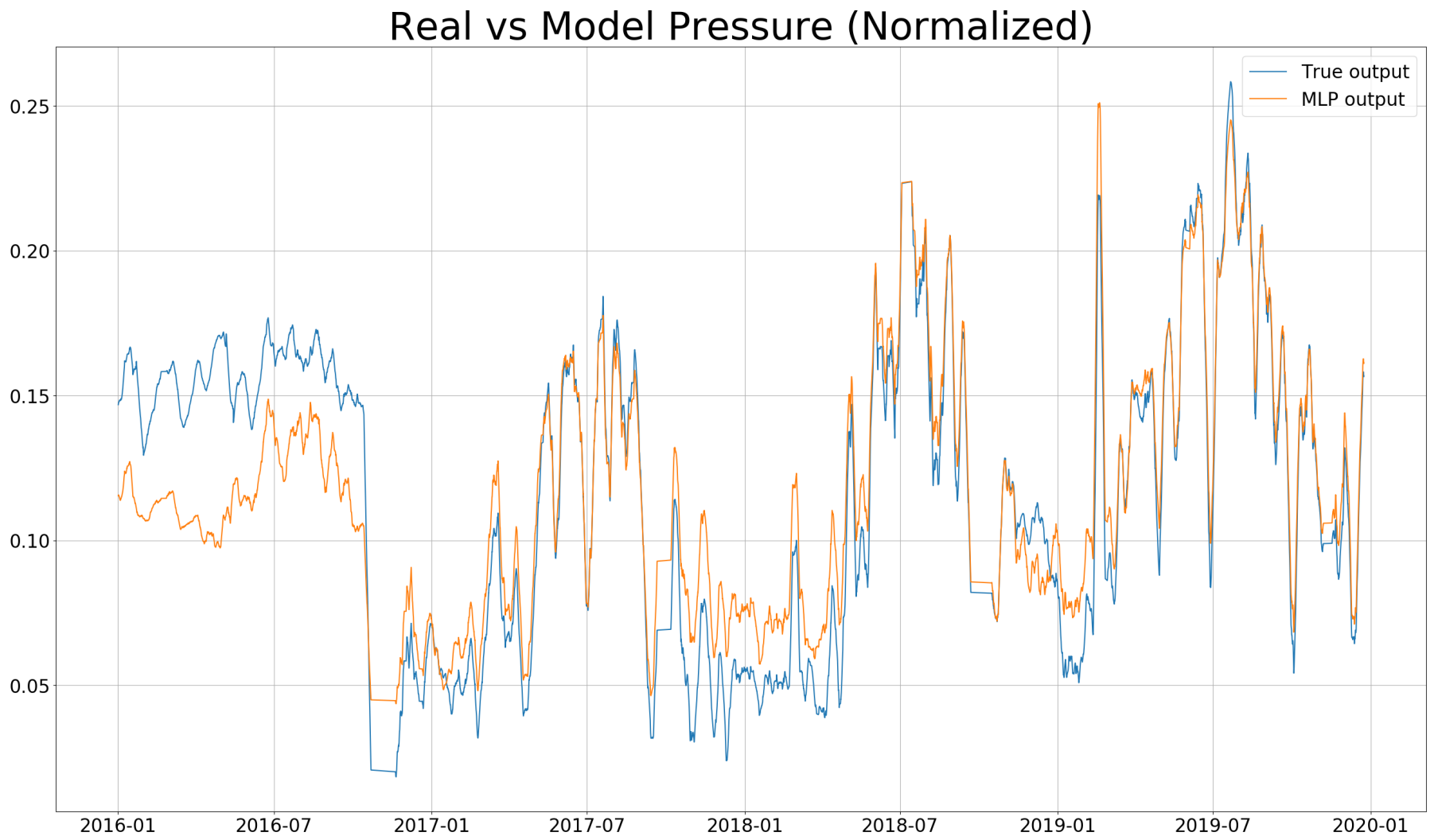

Figure B.6: Pressure output of MLP 3, compared to the true output.

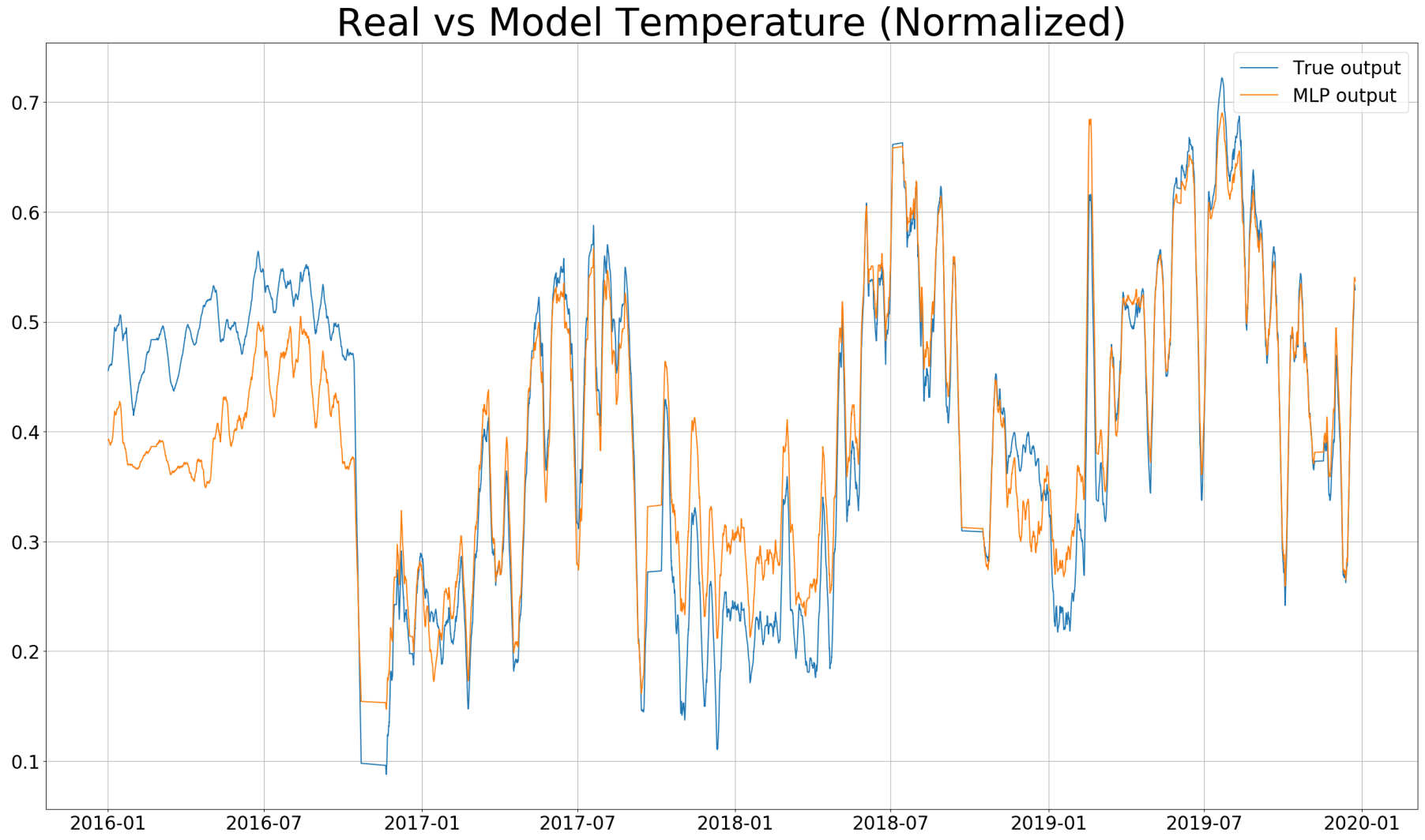

Figure B.7: Temperature output of MLP 3, compared to the true output.

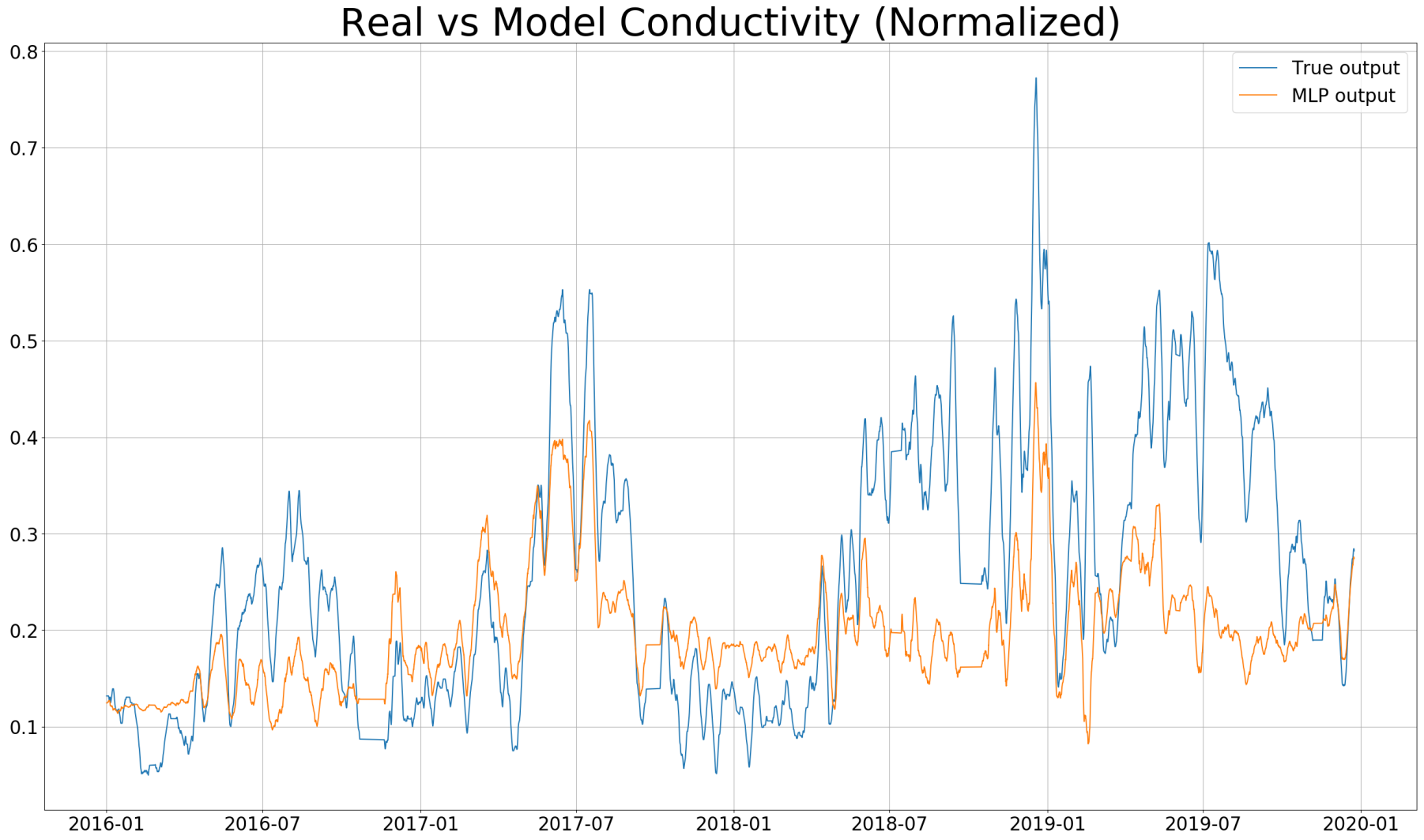

Figure B.8: Conductivity output of MLP 3, compared to the true output.

B.3 LSTM 1

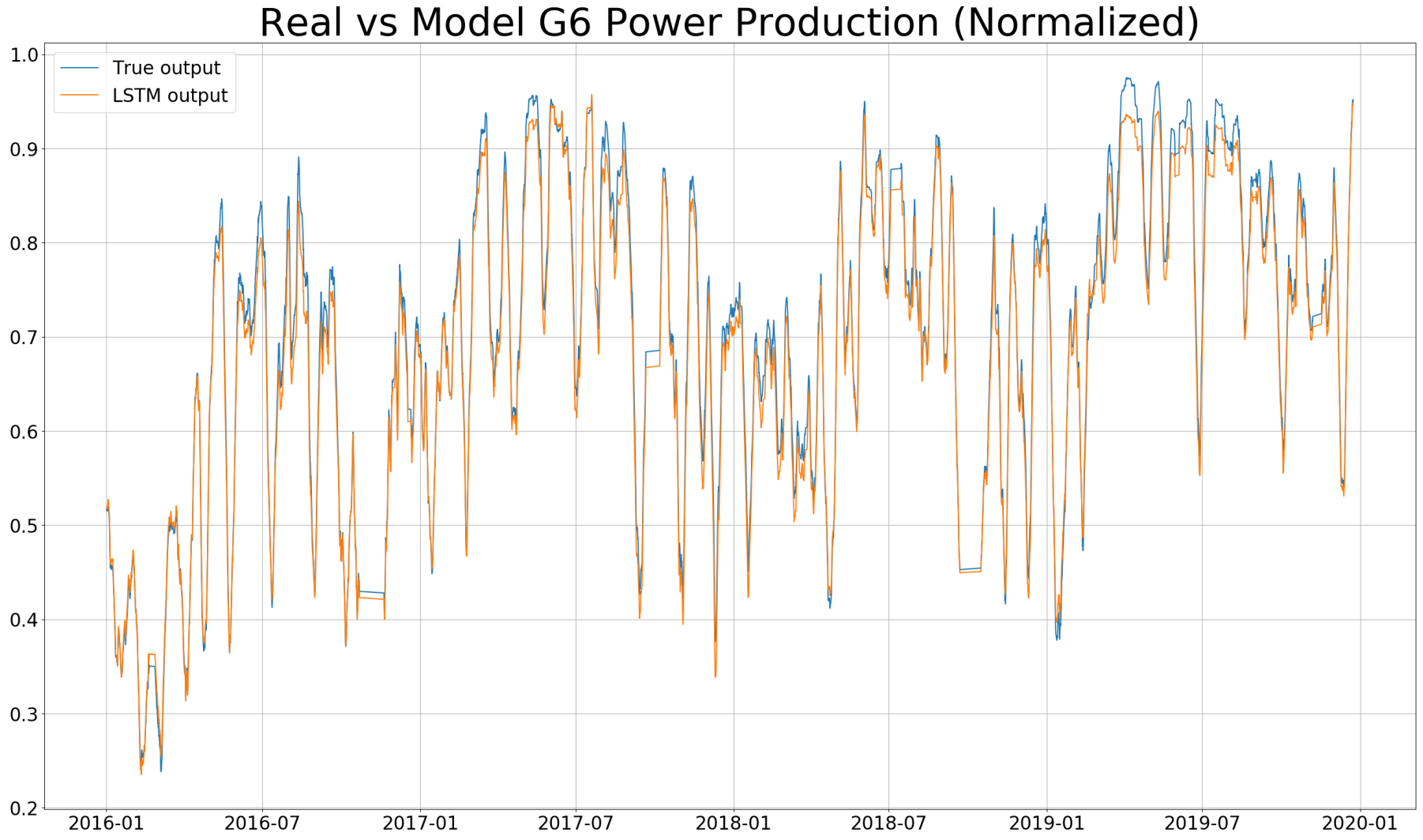

Figure B.9: Power production output of LSTM 1, compared to the true output.

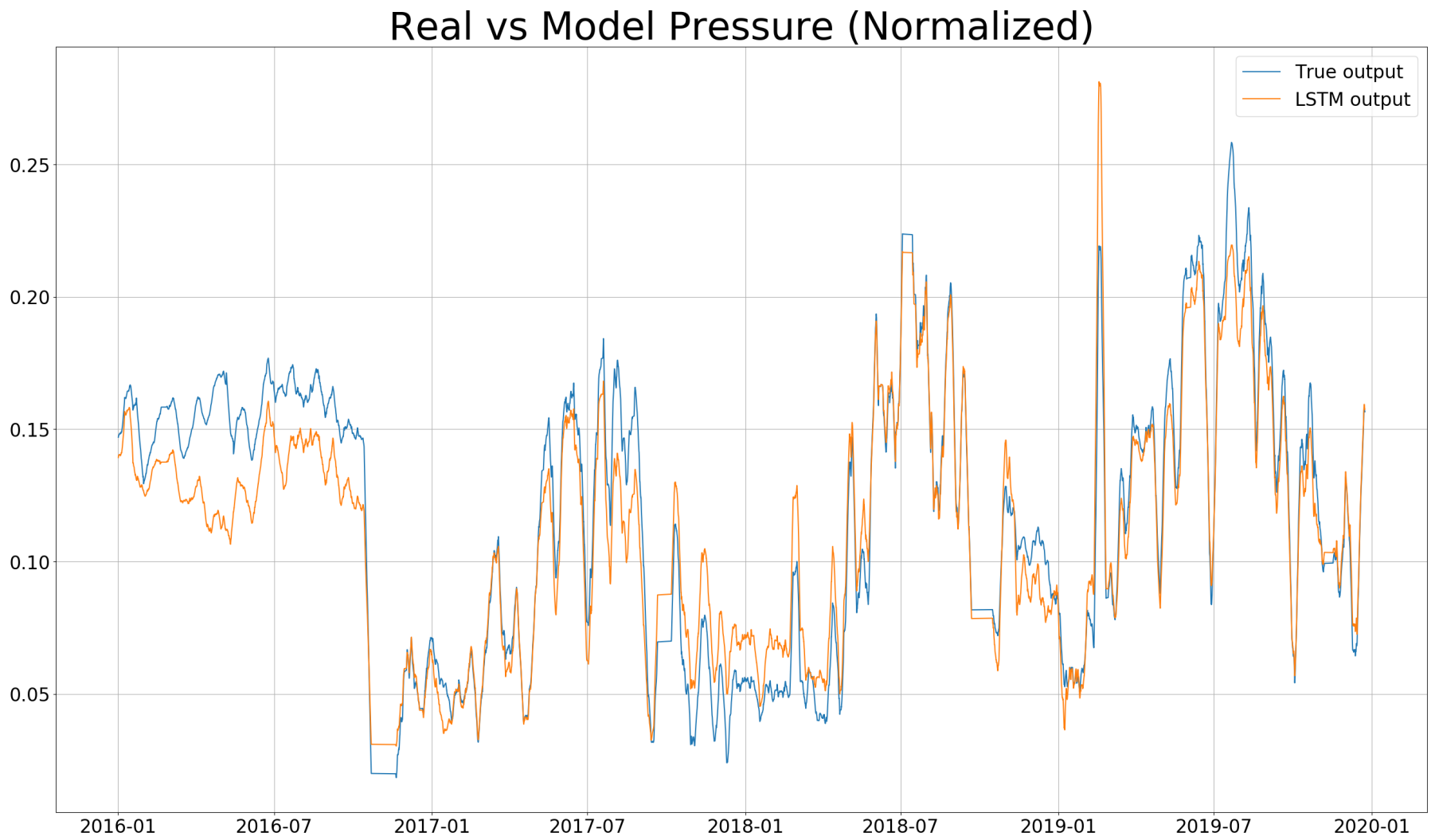

Figure B.10: Pressure output of LSTM 1, compared to the true output.

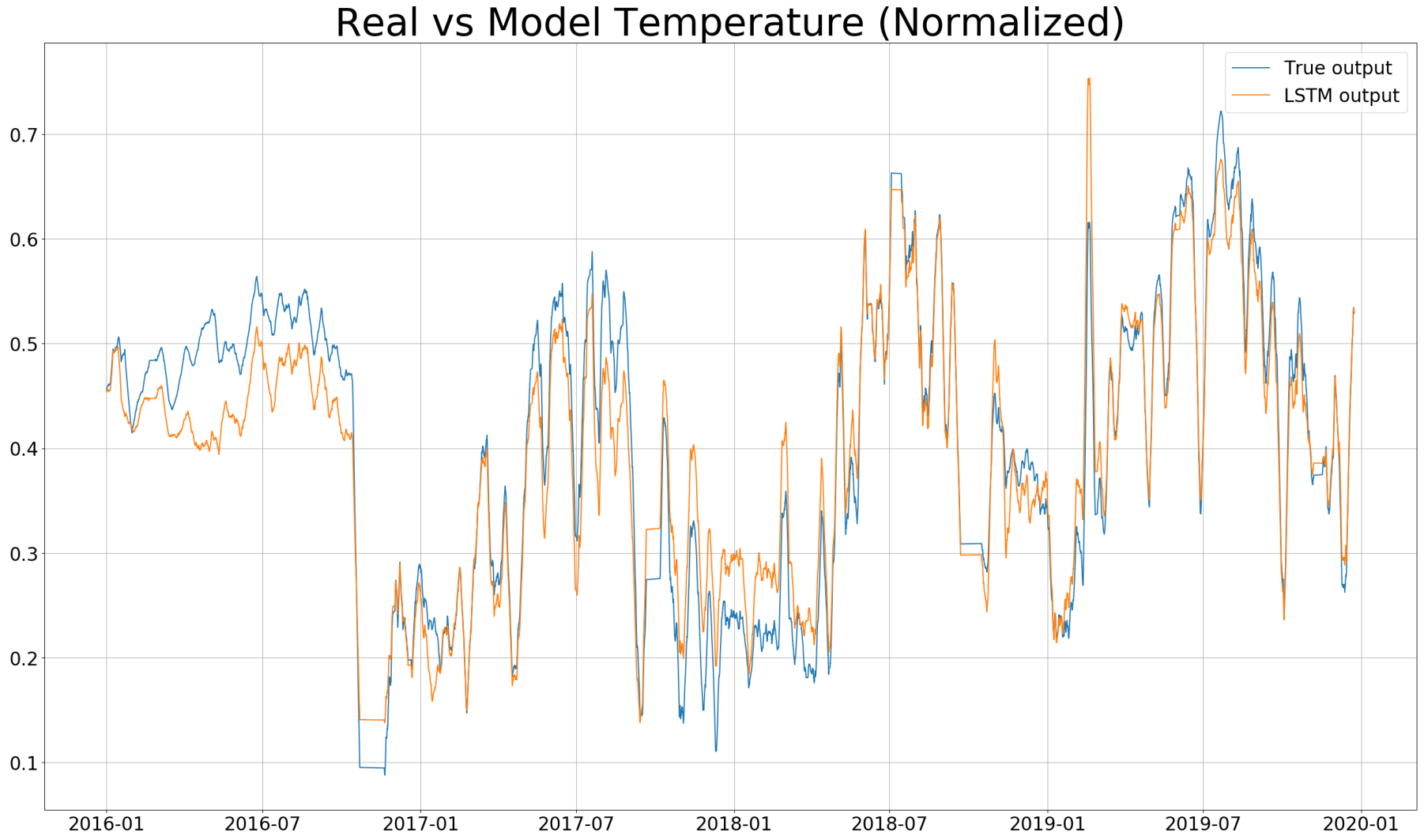

Figure B.11: Temperature output of LSTM 1, compared to the true output.

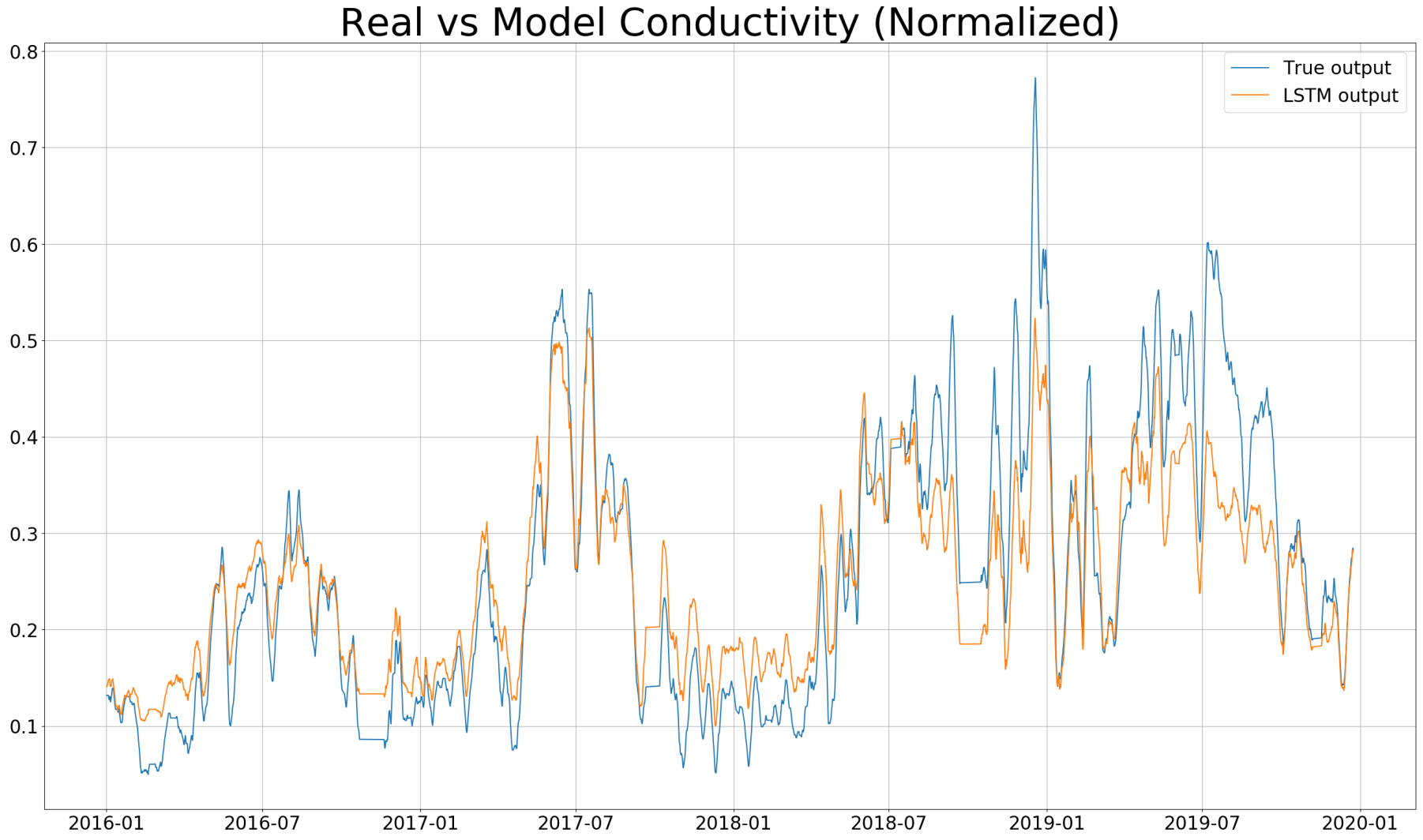

Figure B.12: Conductivity output of LSTM 1, compared to the true output.

B.4 LSTM 2

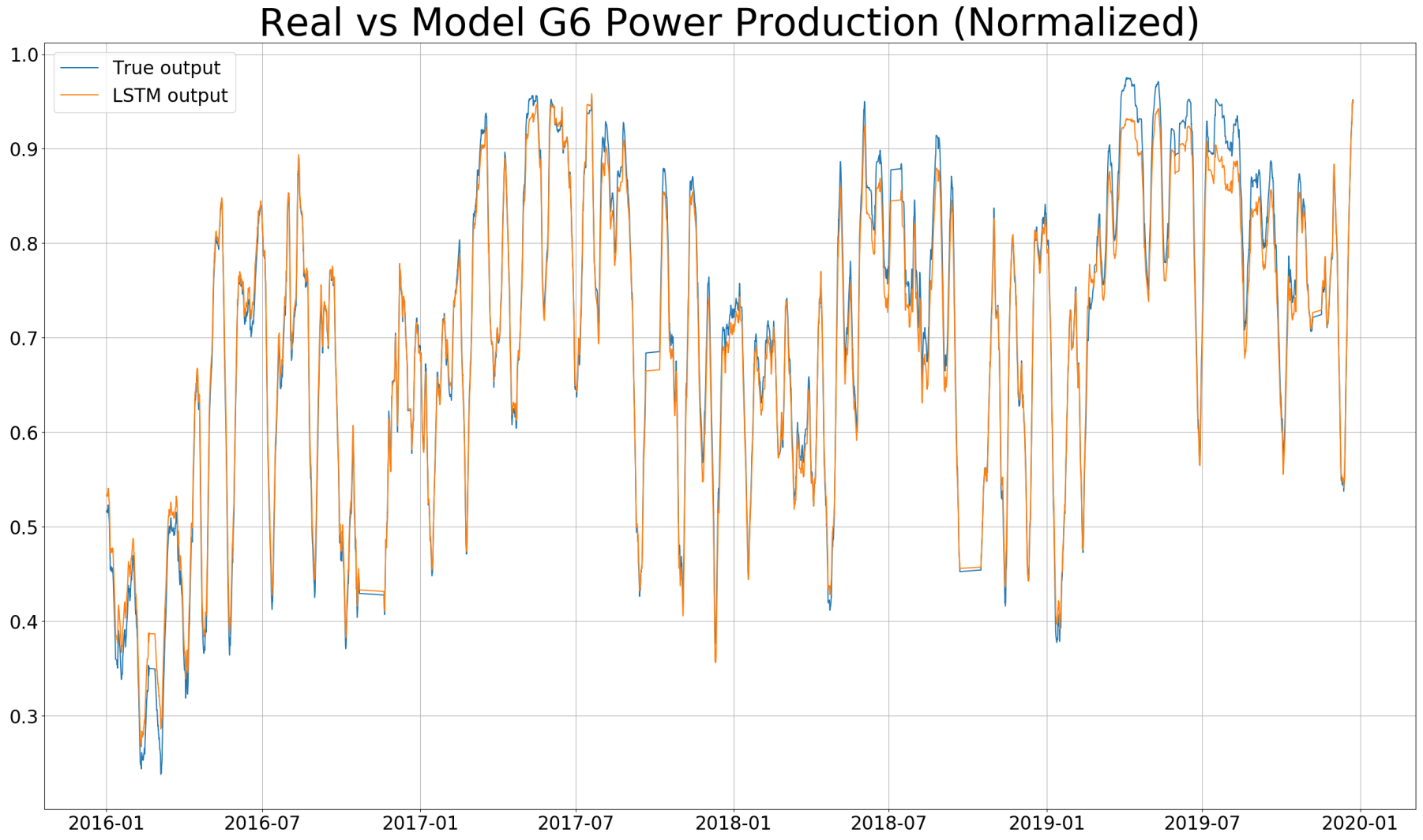

Figure B.13: Power production output of LSTM 2, compared to the true output.

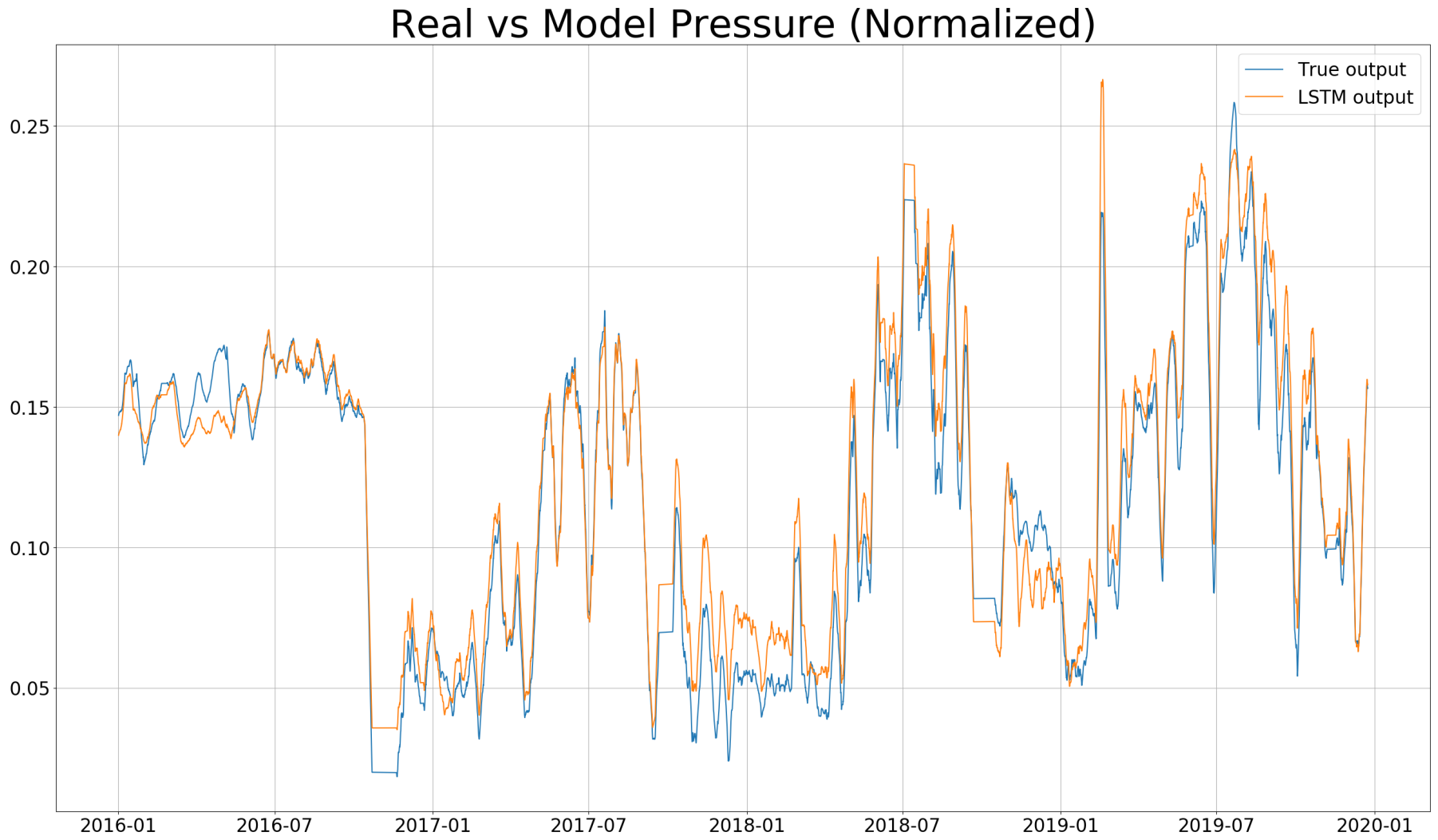

Figure B.14: Pressure output of LSTM 2, compared to the true output.

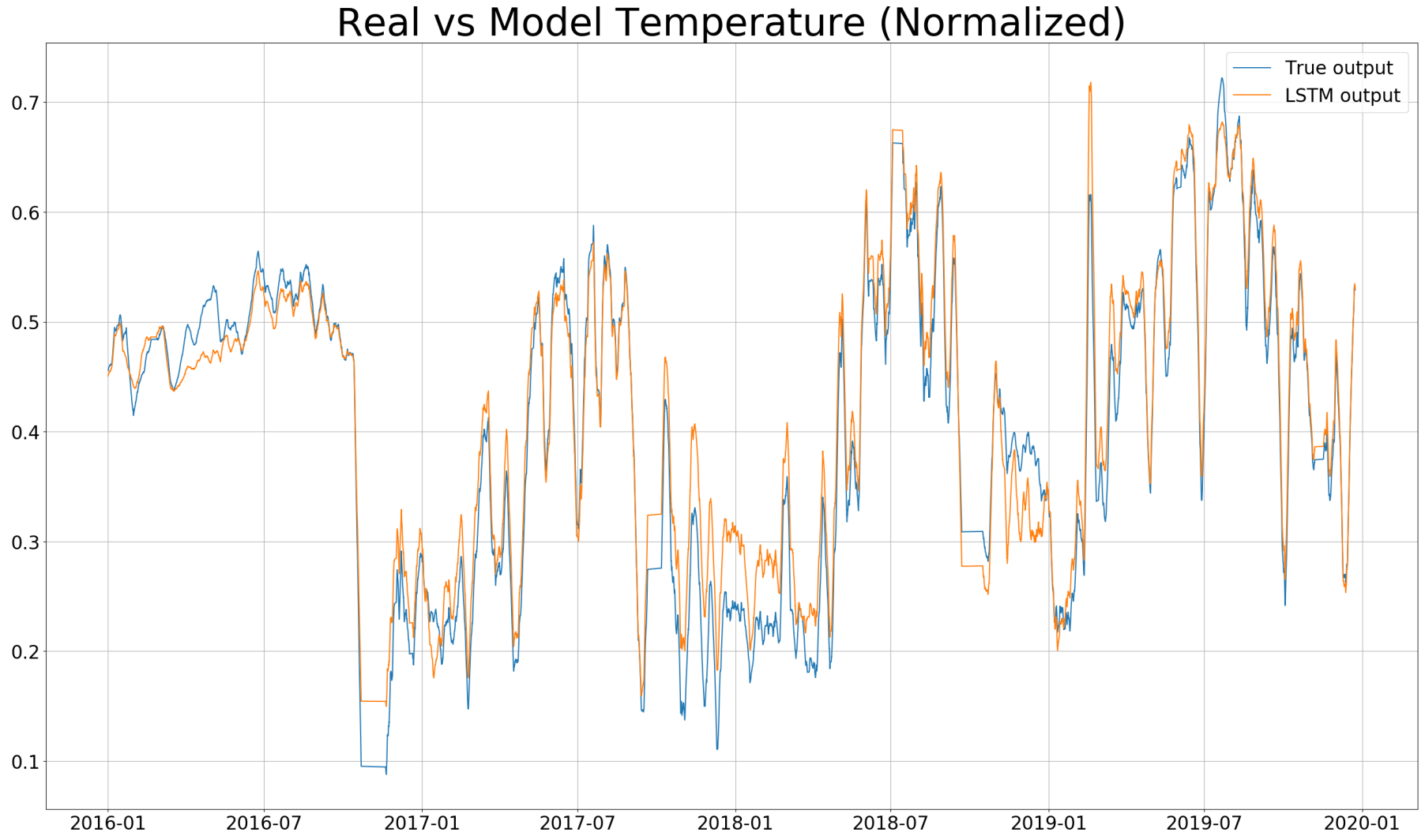

Figure B.15: Temperature output of LSTM 2, compared to the true output.

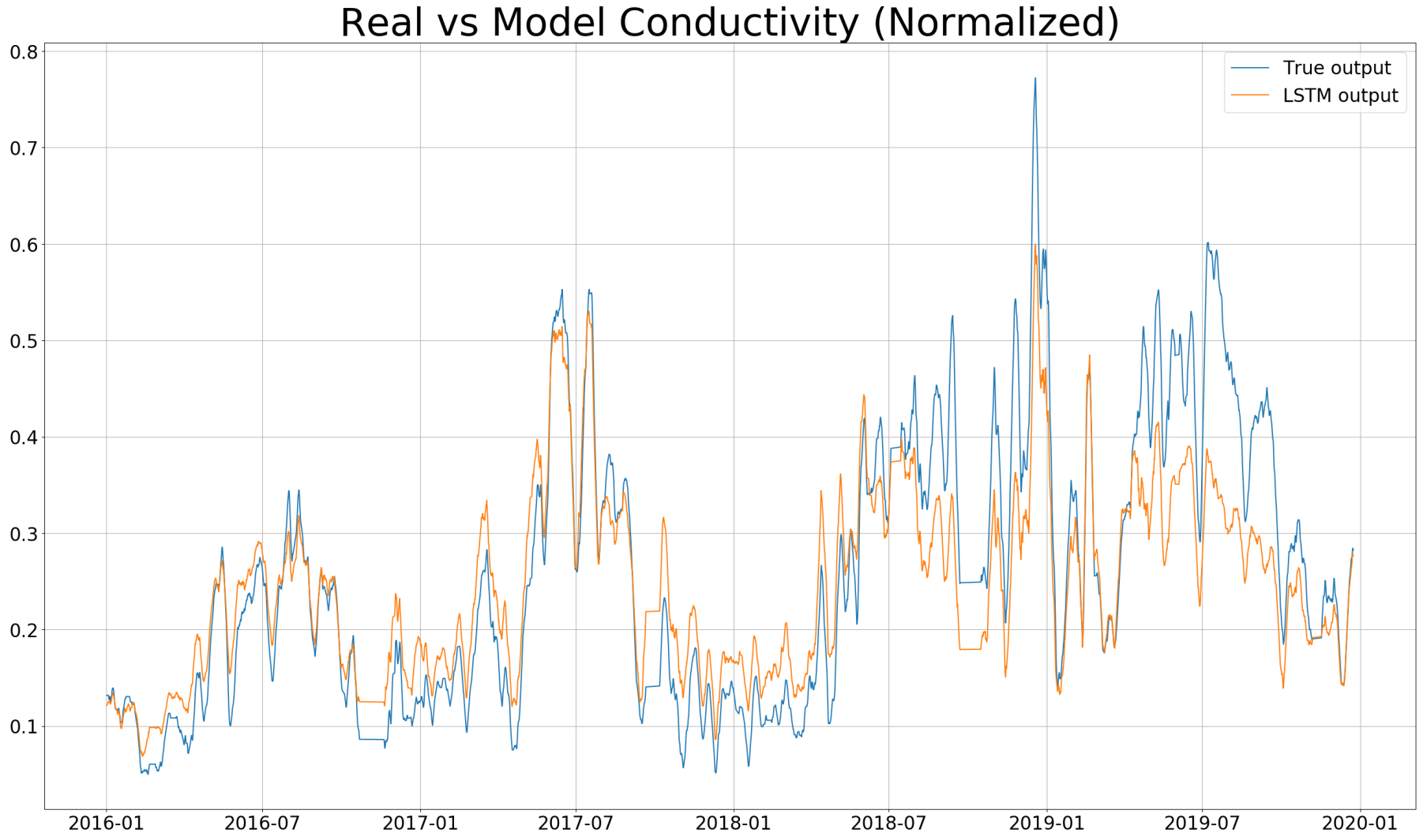

Figure B.16: Conductivity output of LSTM 2, compared to the true output.

B.5 LSTM 3

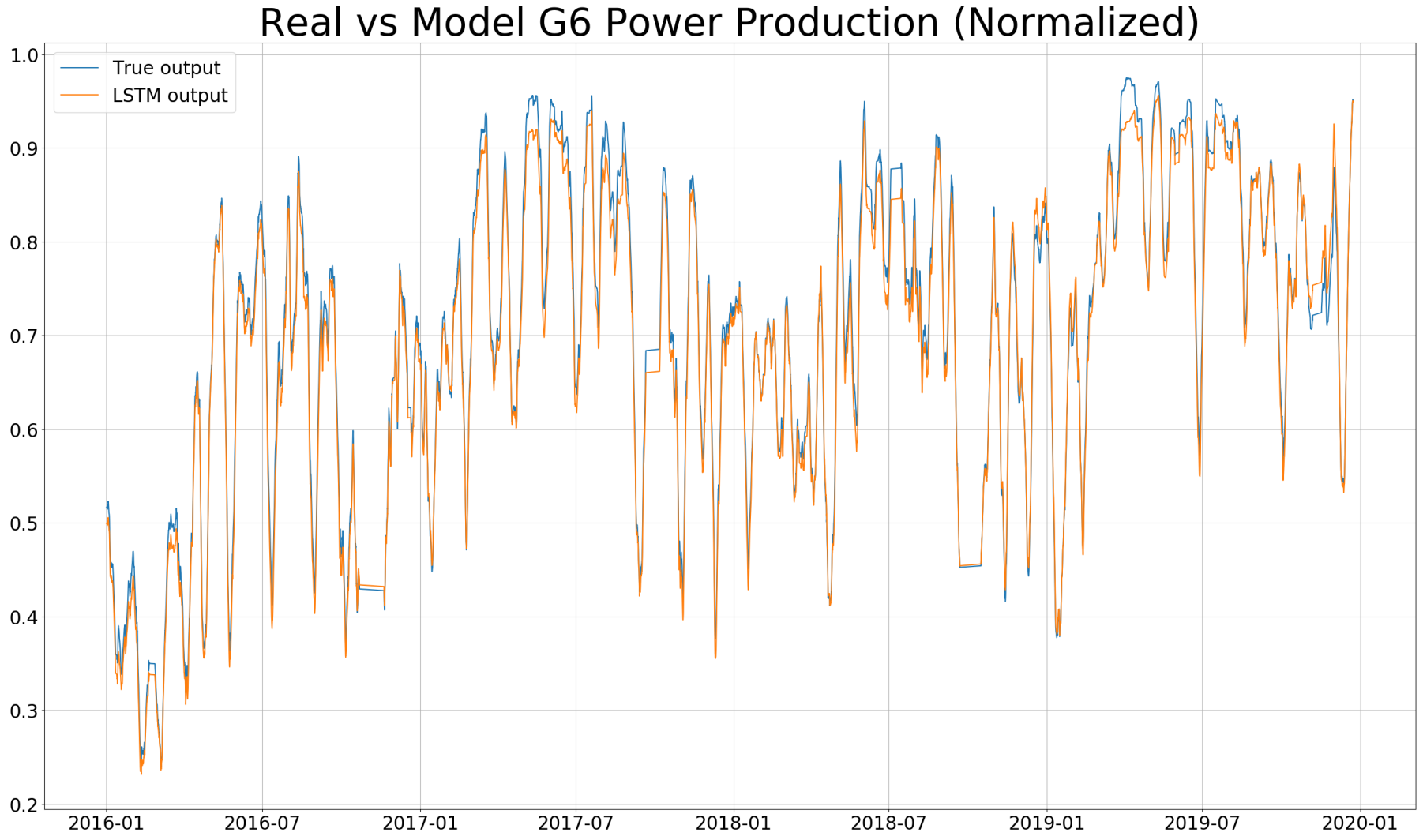

Figure B.17: Power production output of LSTM 3, compared to the true output.

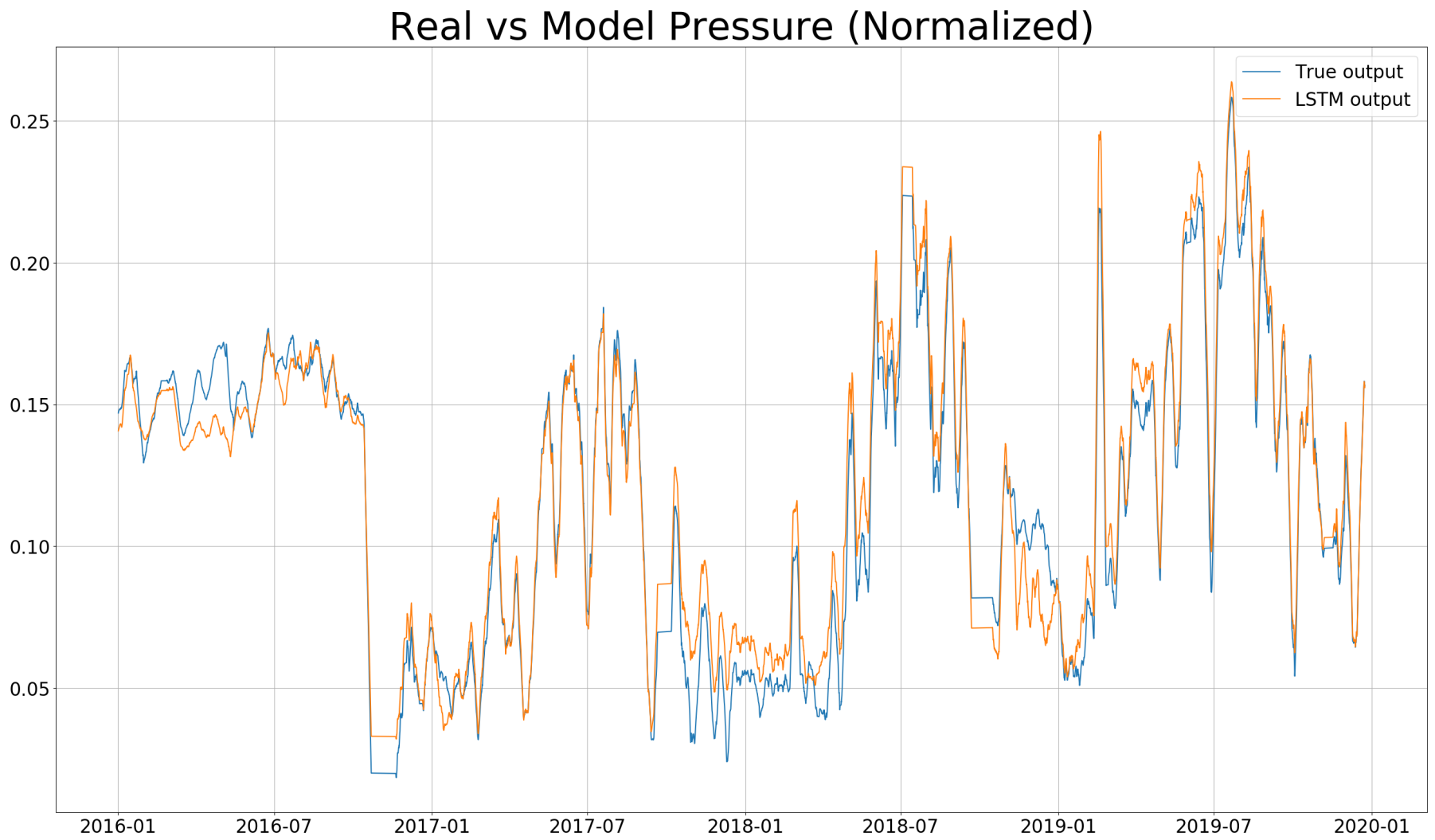

Figure B.18: Pressure output of LSTM 3, compared to the true output.

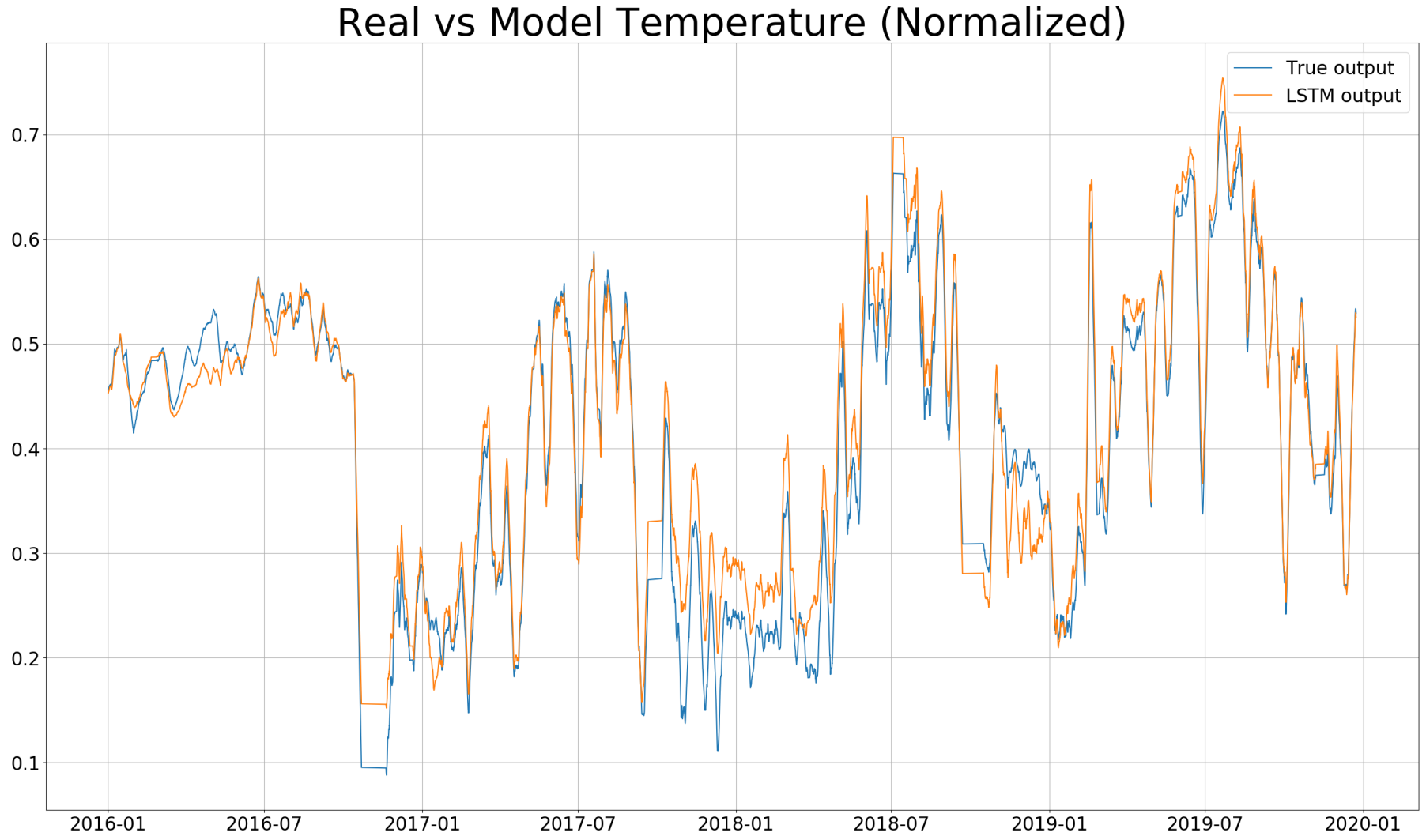

Figure B.19: Temperature output of LSTM 3, compared to the true output.

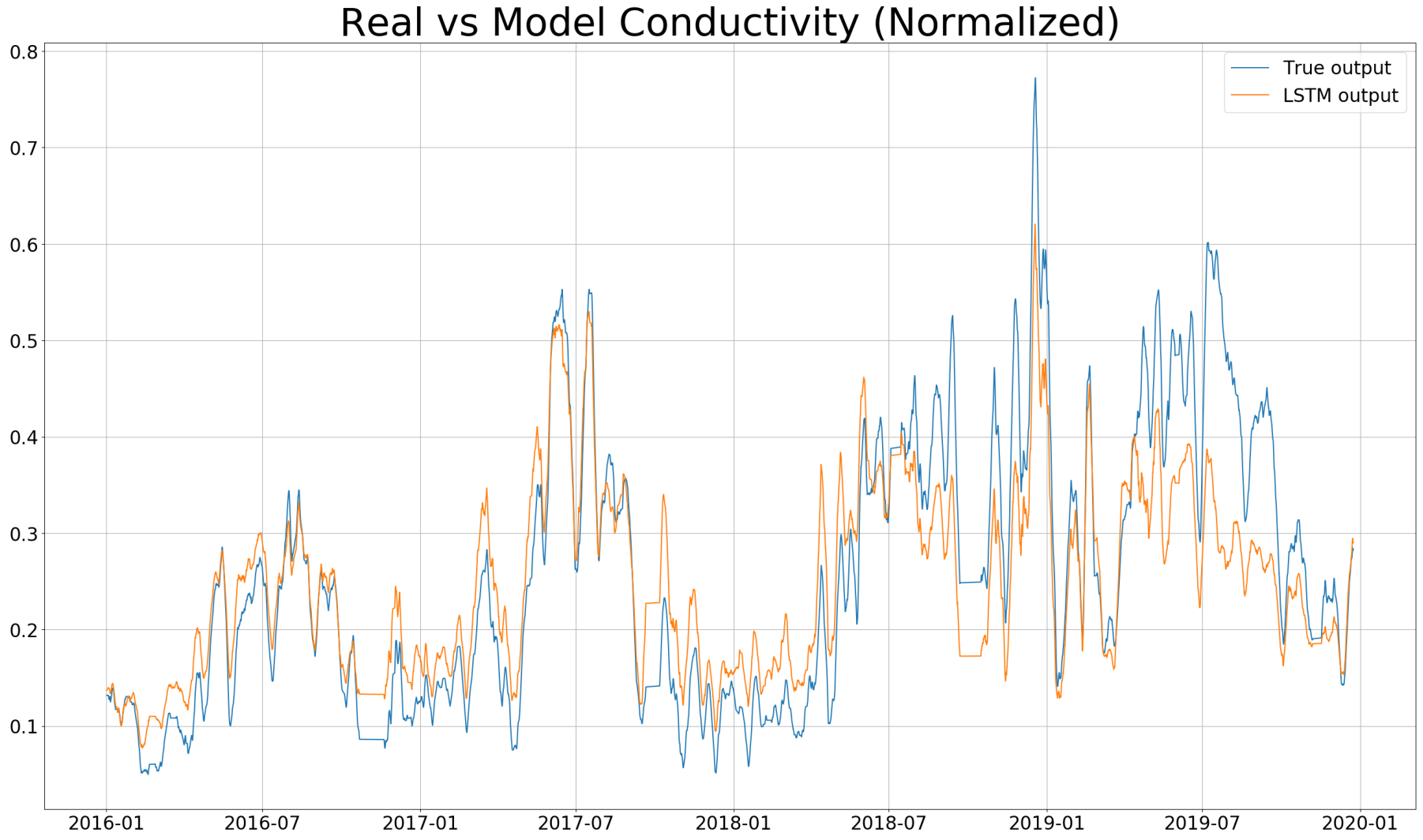

Figure B.20: Conductivity output of LSTM 3, compared to the true output.

B.6 LSTM 4

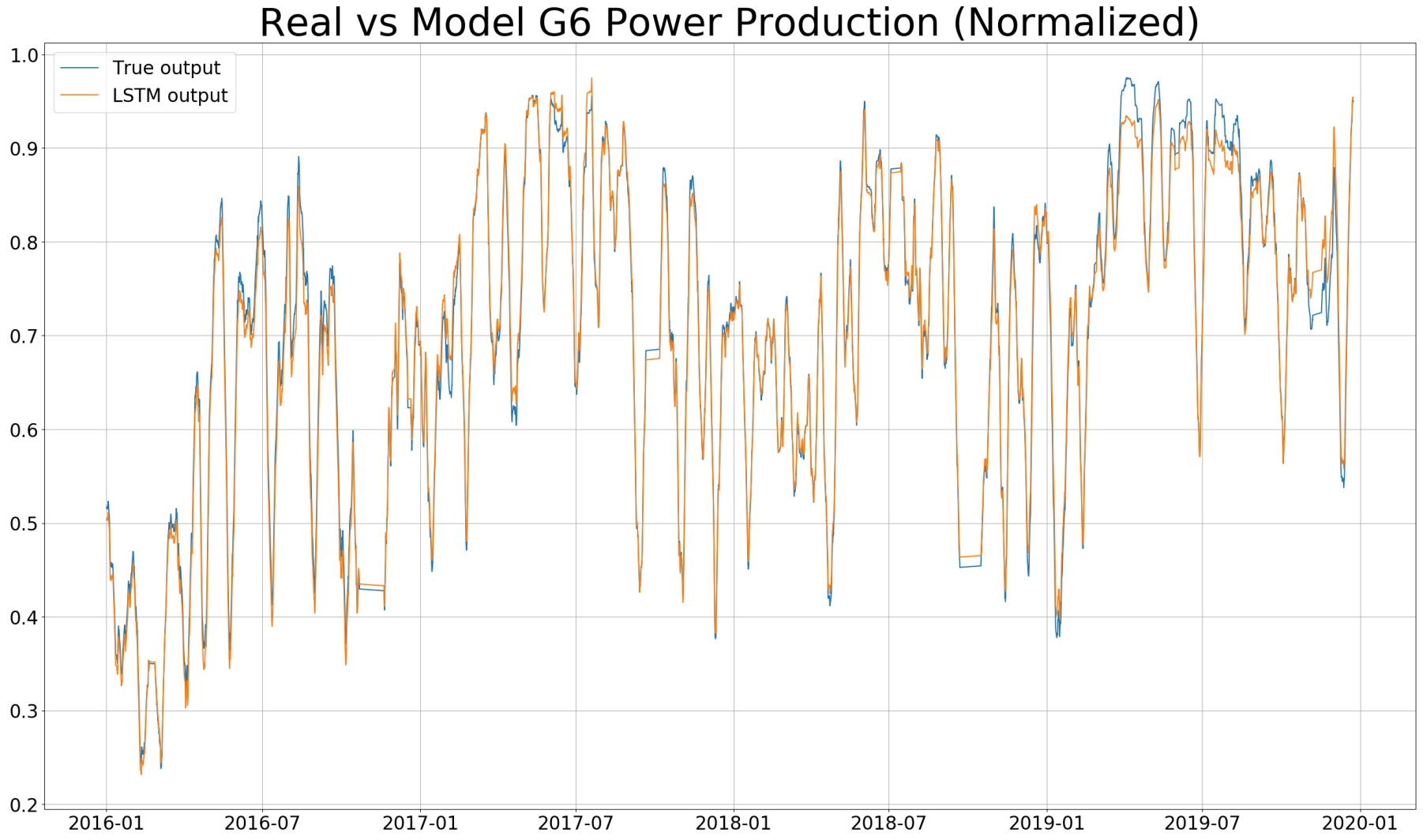

Figure B.21: Power production output of LSTM 4, compared to the true output.

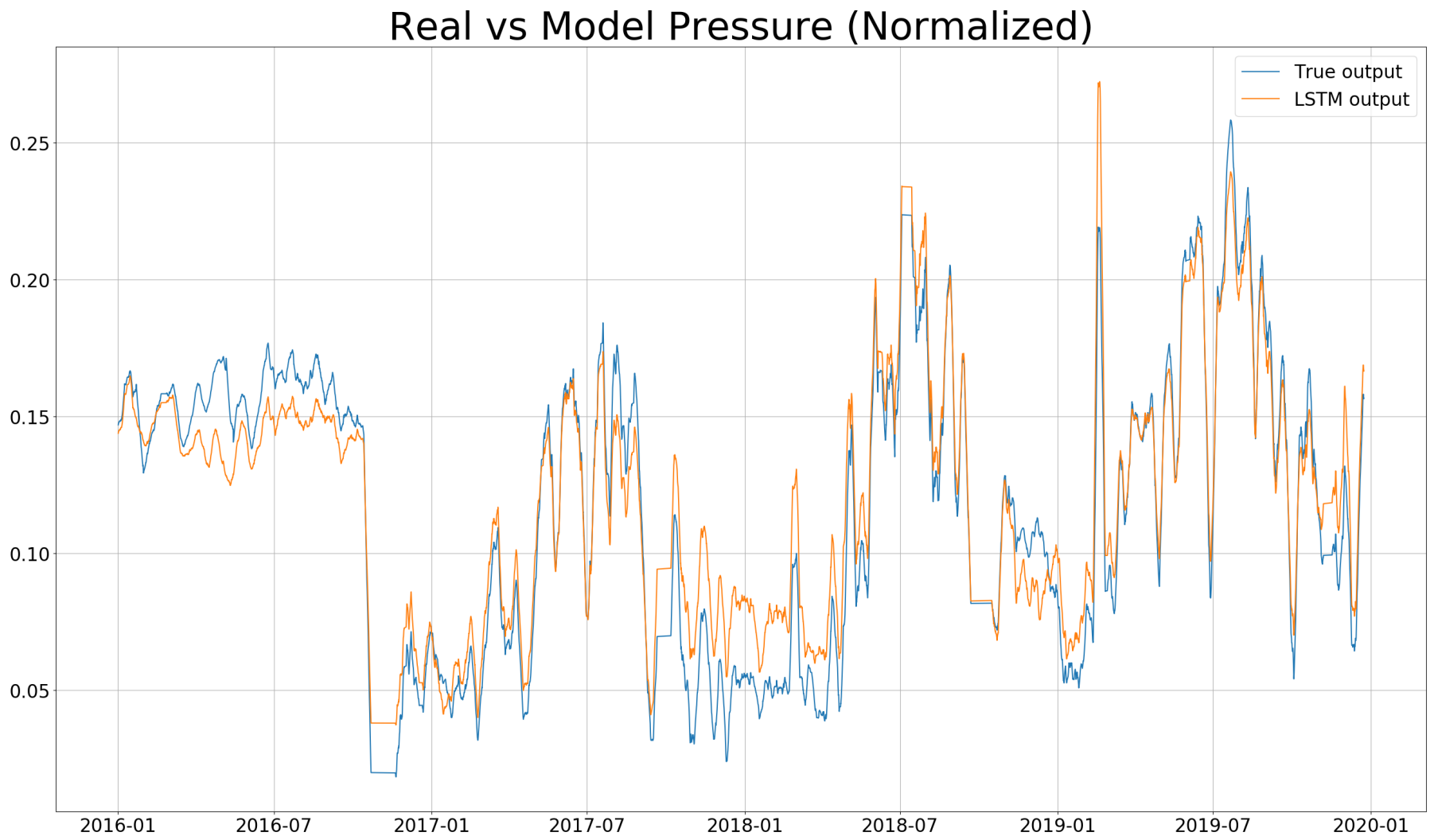

Figure B.22: Pressure output of LSTM 4, compared to the true output.

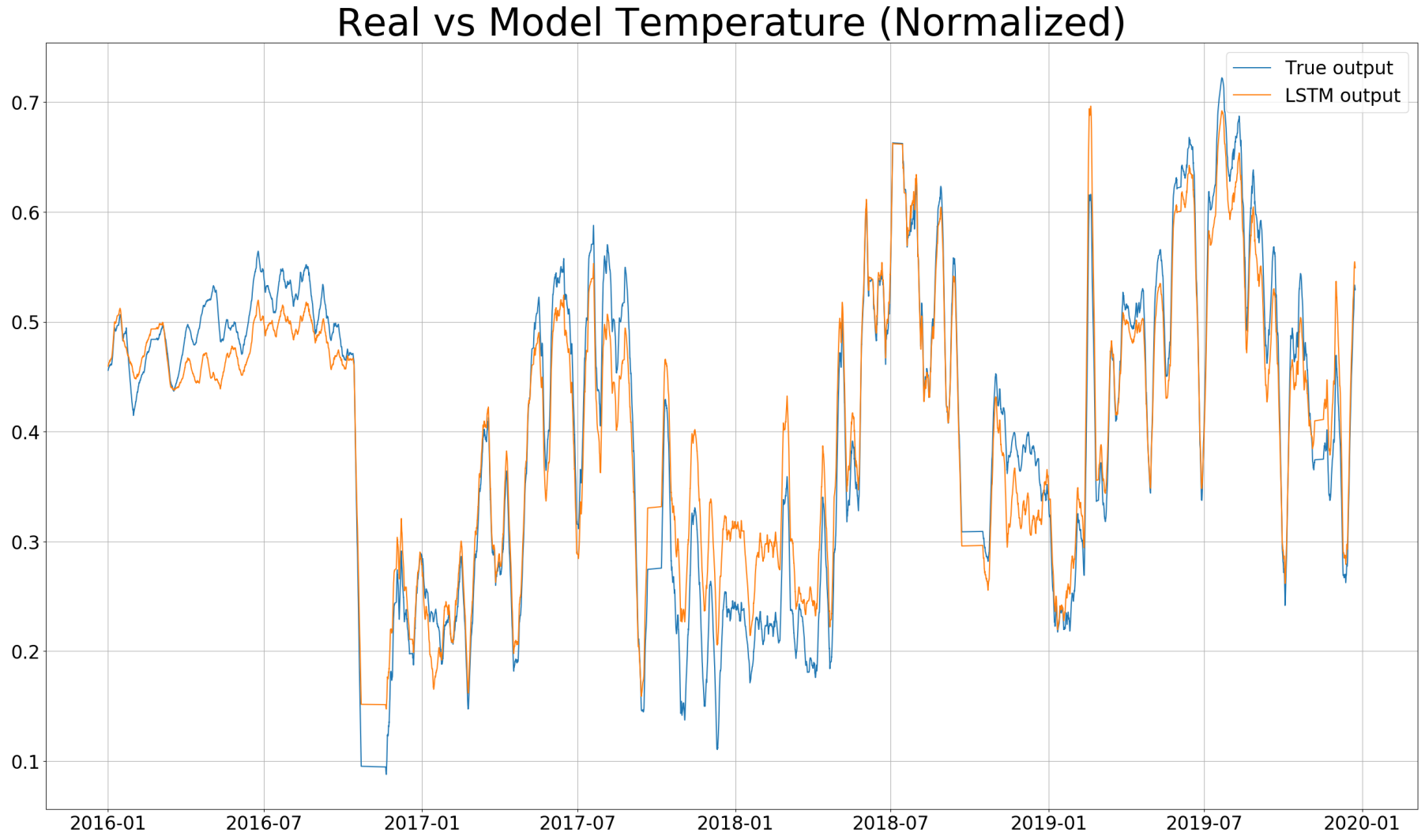

Figure B.23: Temperature output of LSTM 4, compared to the true output.

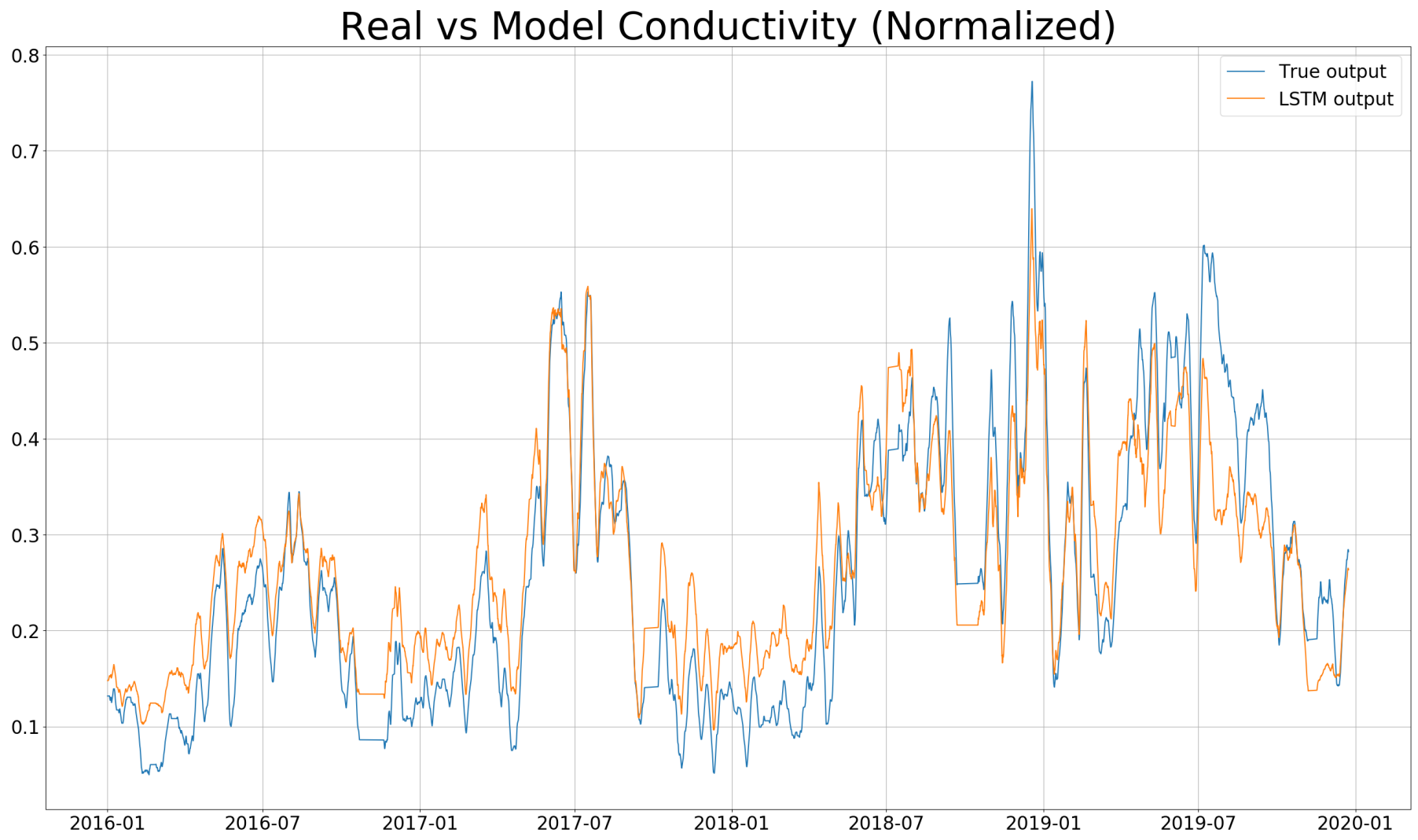

Figure B.24: Conductivity output of LSTM 4, compared to the true output.

B.7 LSTM 5

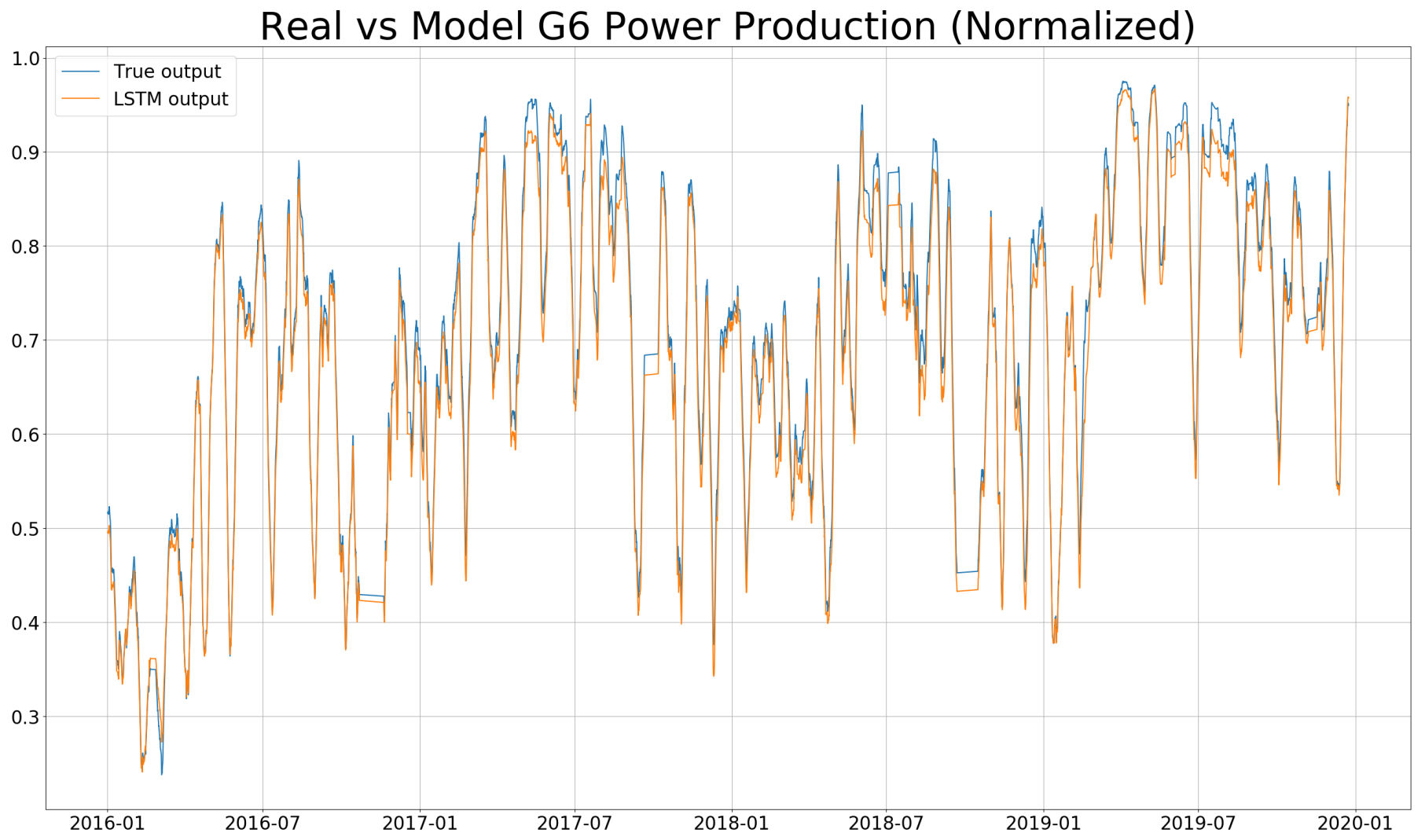

Figure B.25: Power production output of LSTM 5, compared to the true output.

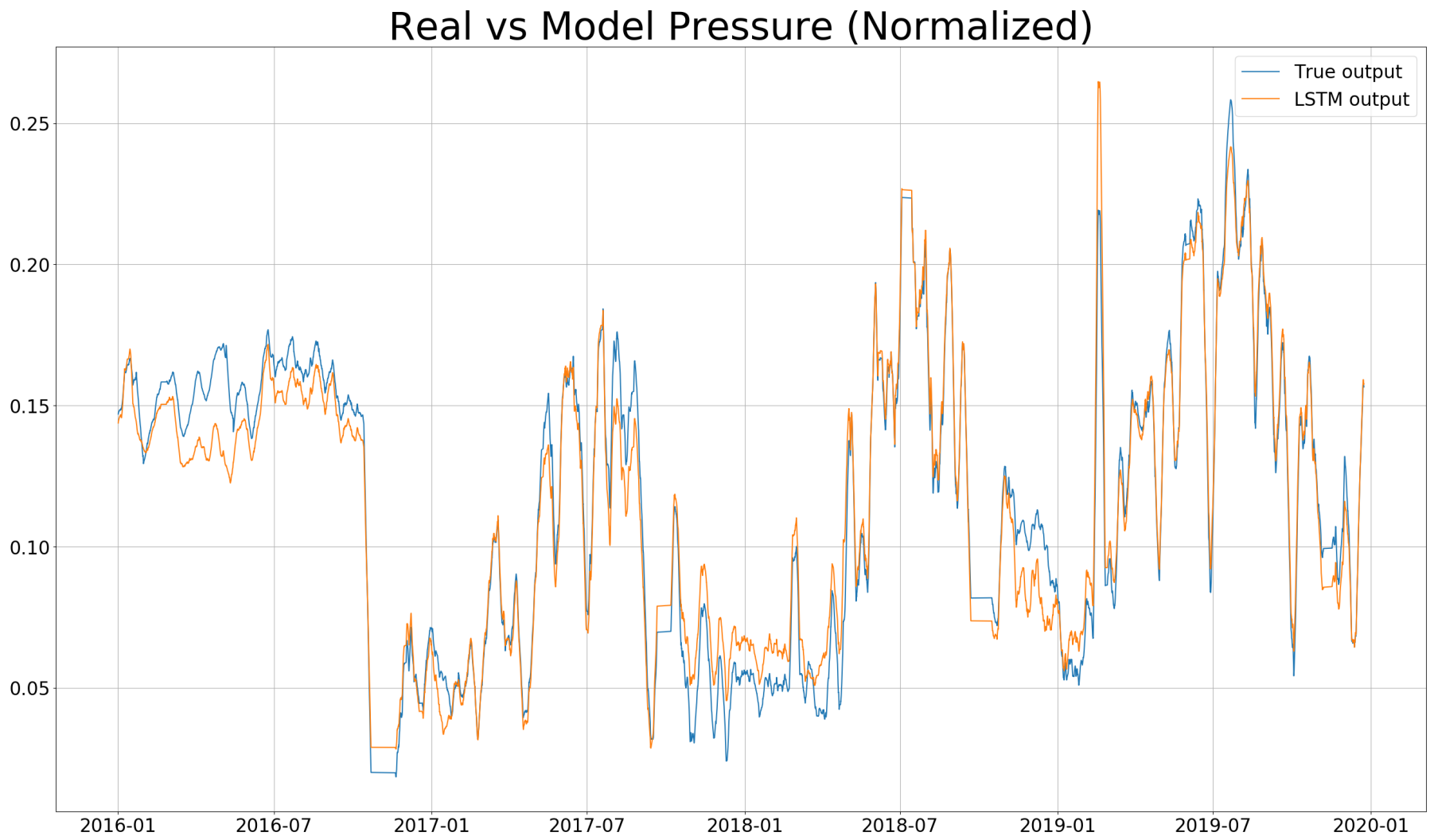

Figure B.26: Pressure output of LSTM 5, compared to the true output.

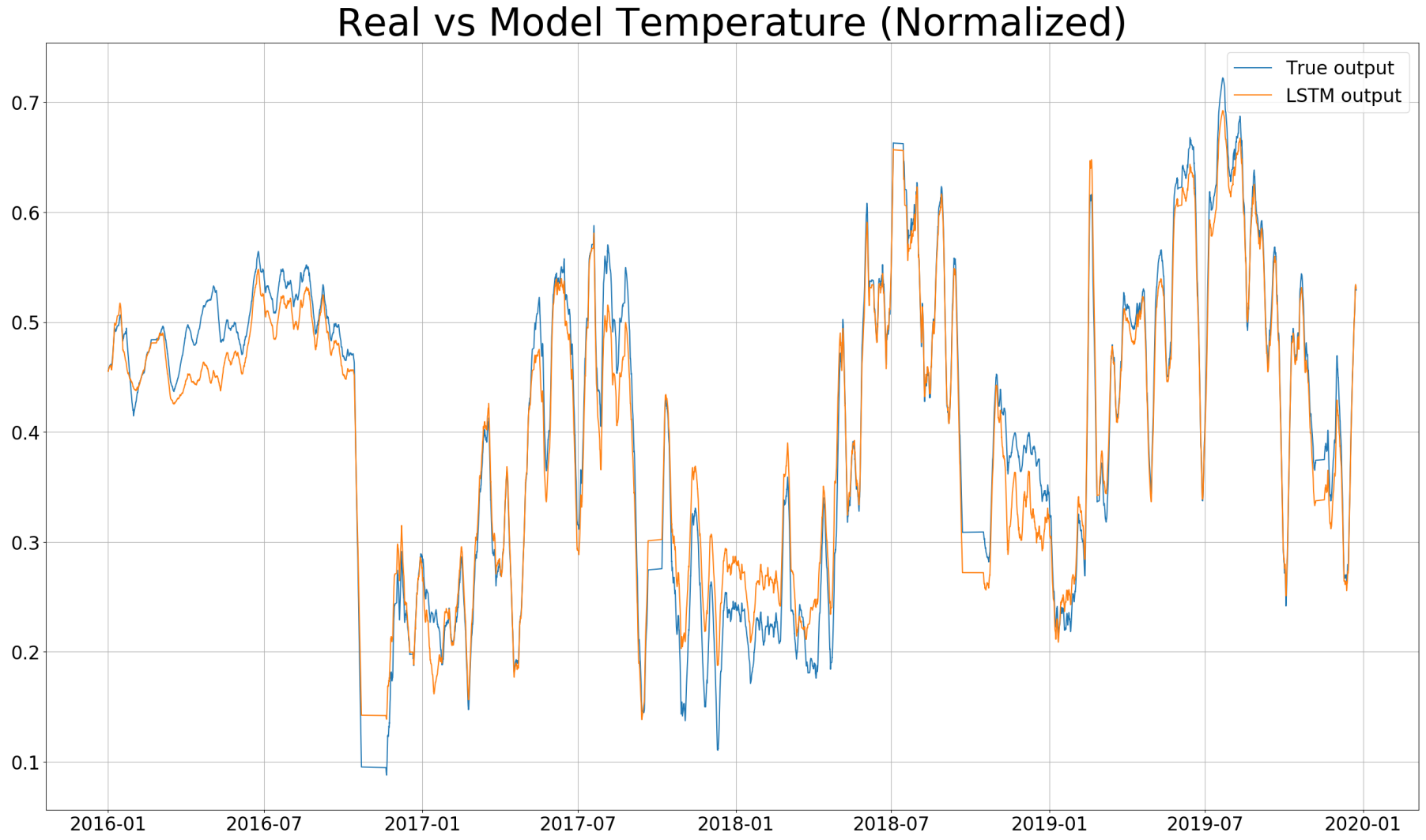

Figure B.27: Temperature output of LSTM 5, compared to the true output.

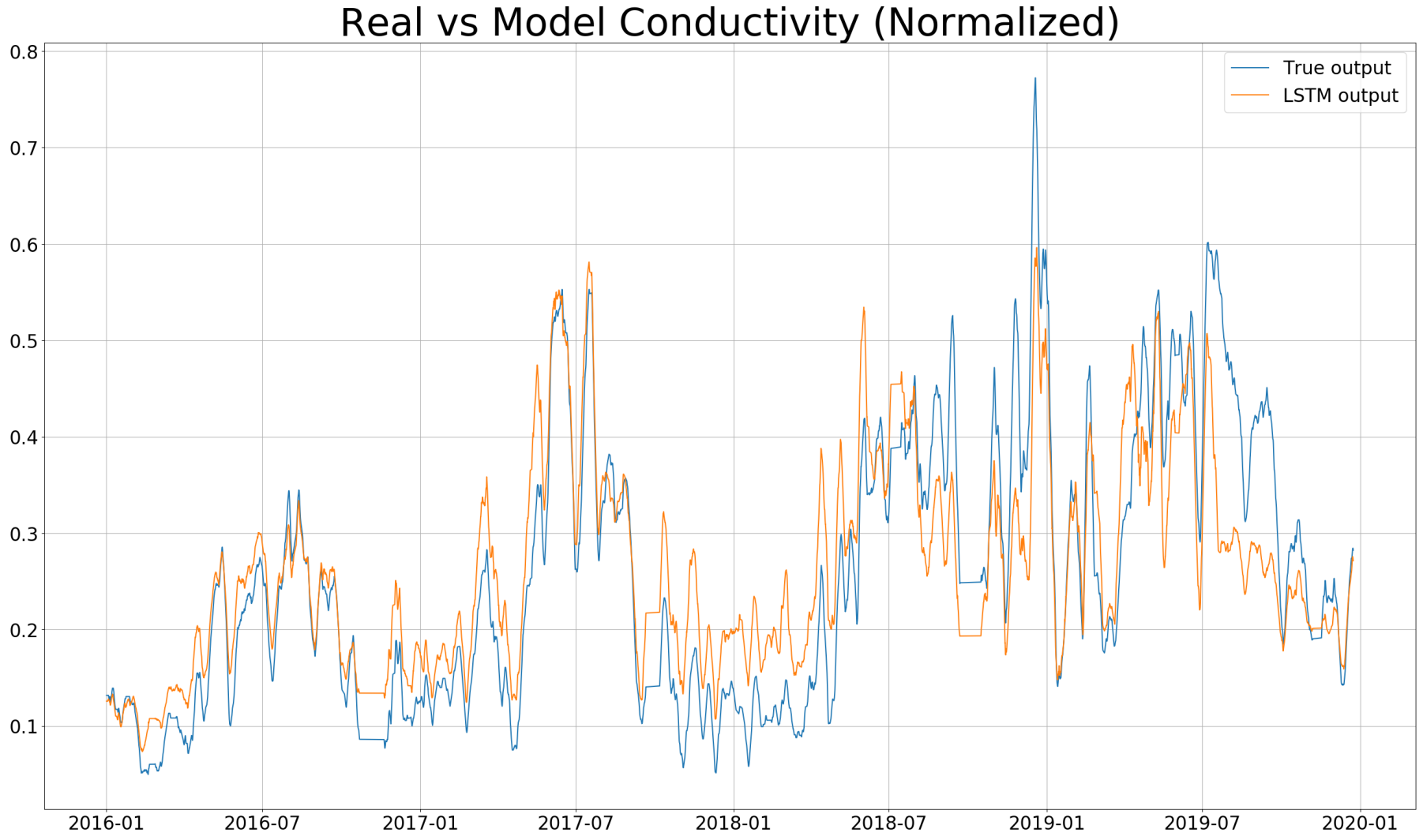

Figure B.28: Conductivity output of LSTM 5, compared to the true output.

B.8 LSTM 6

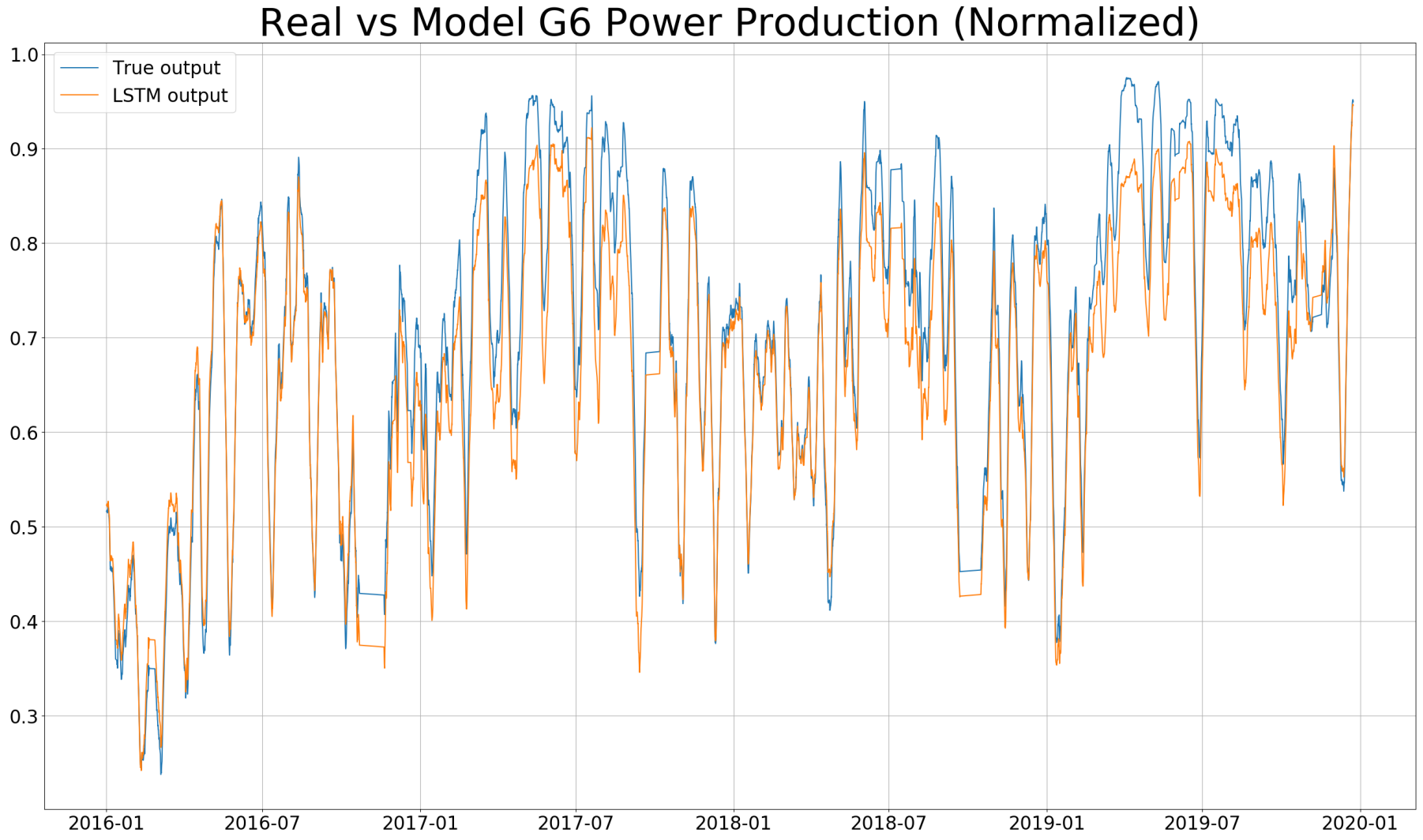

Figure B.29: Power production output of LSTM 6, compared to the true output.

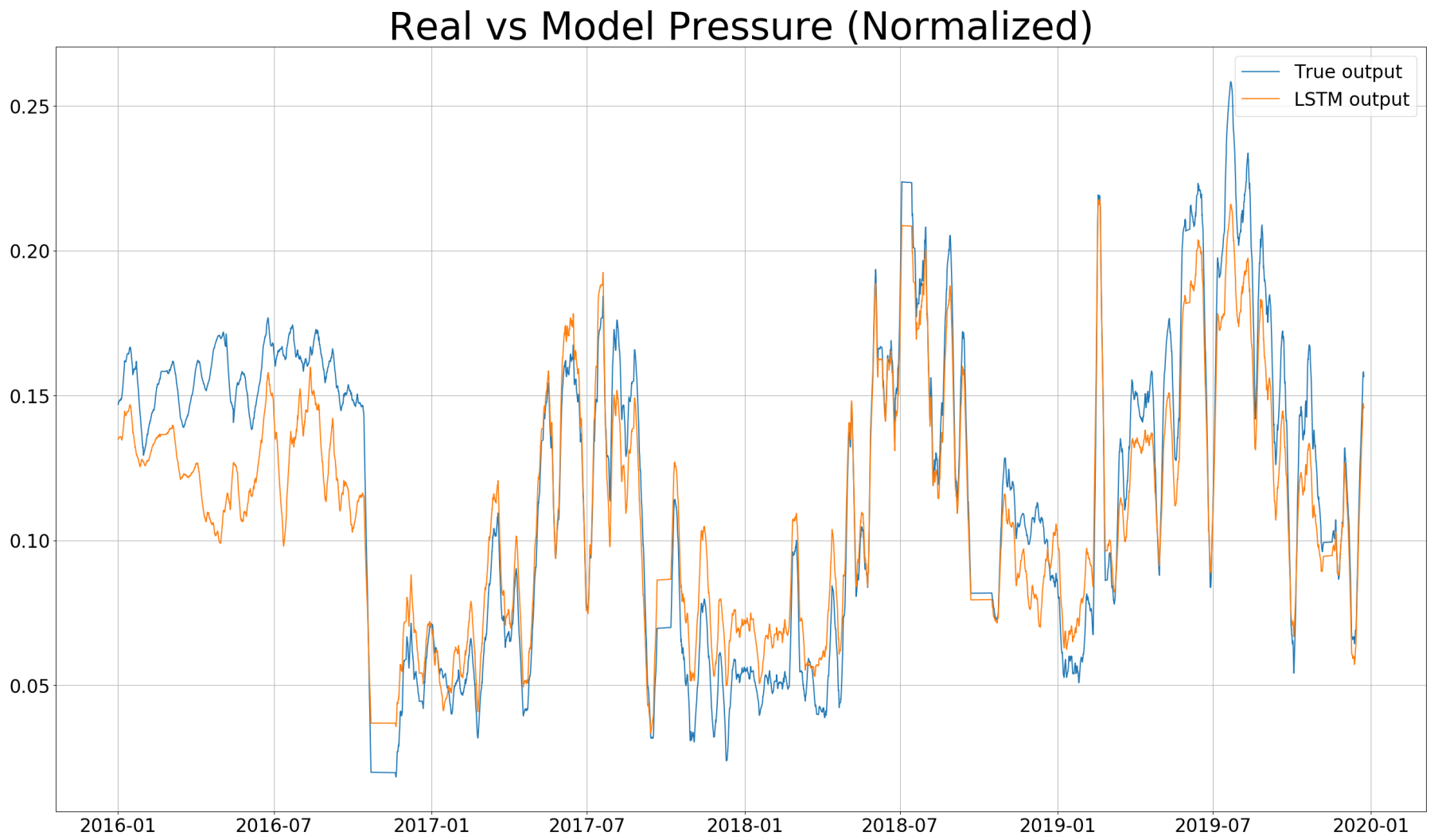

Figure B.30: Pressure output of LSTM 6, compared to the true output.

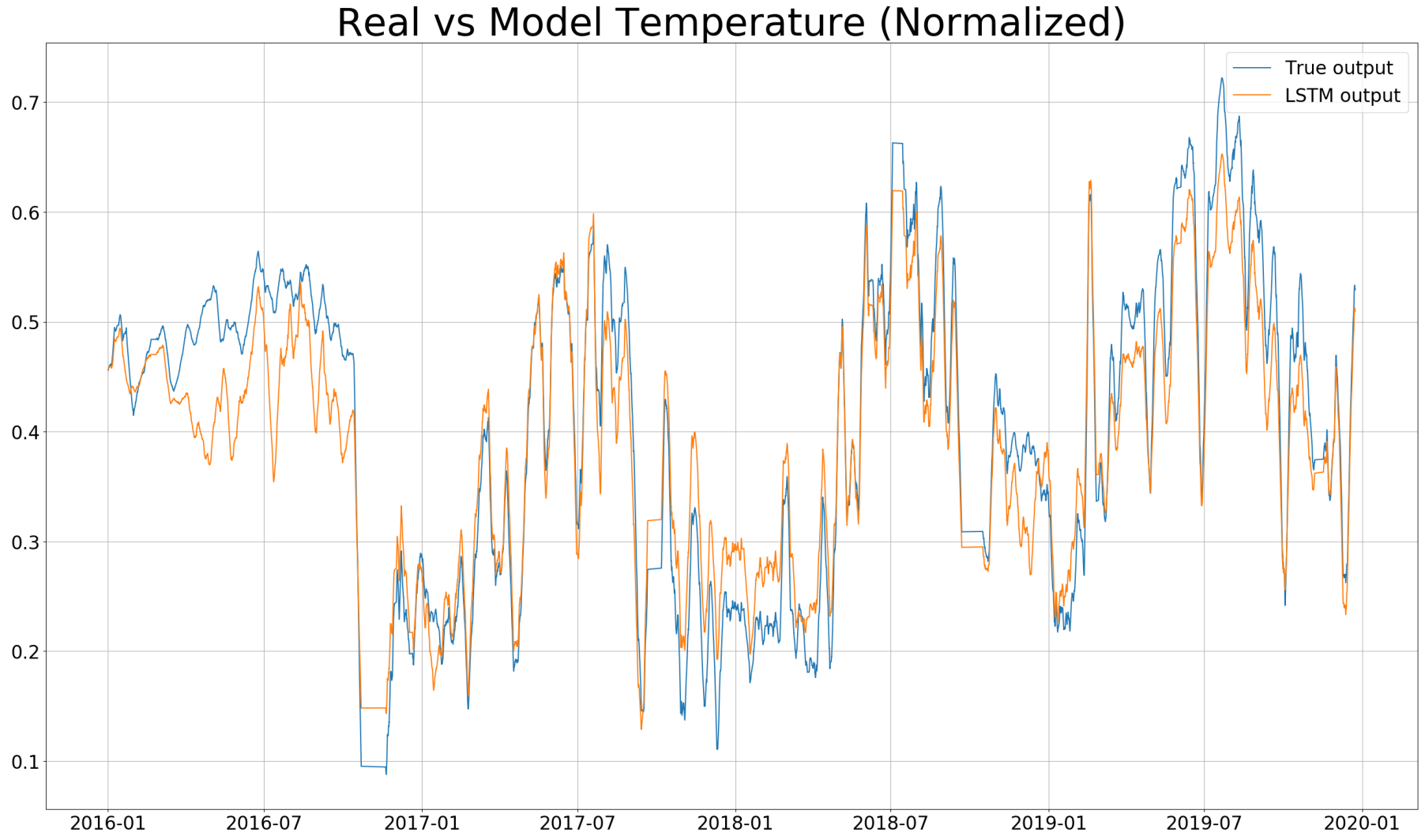

Figure B.31: Temperature output of LSTM 6, compared to the true output.

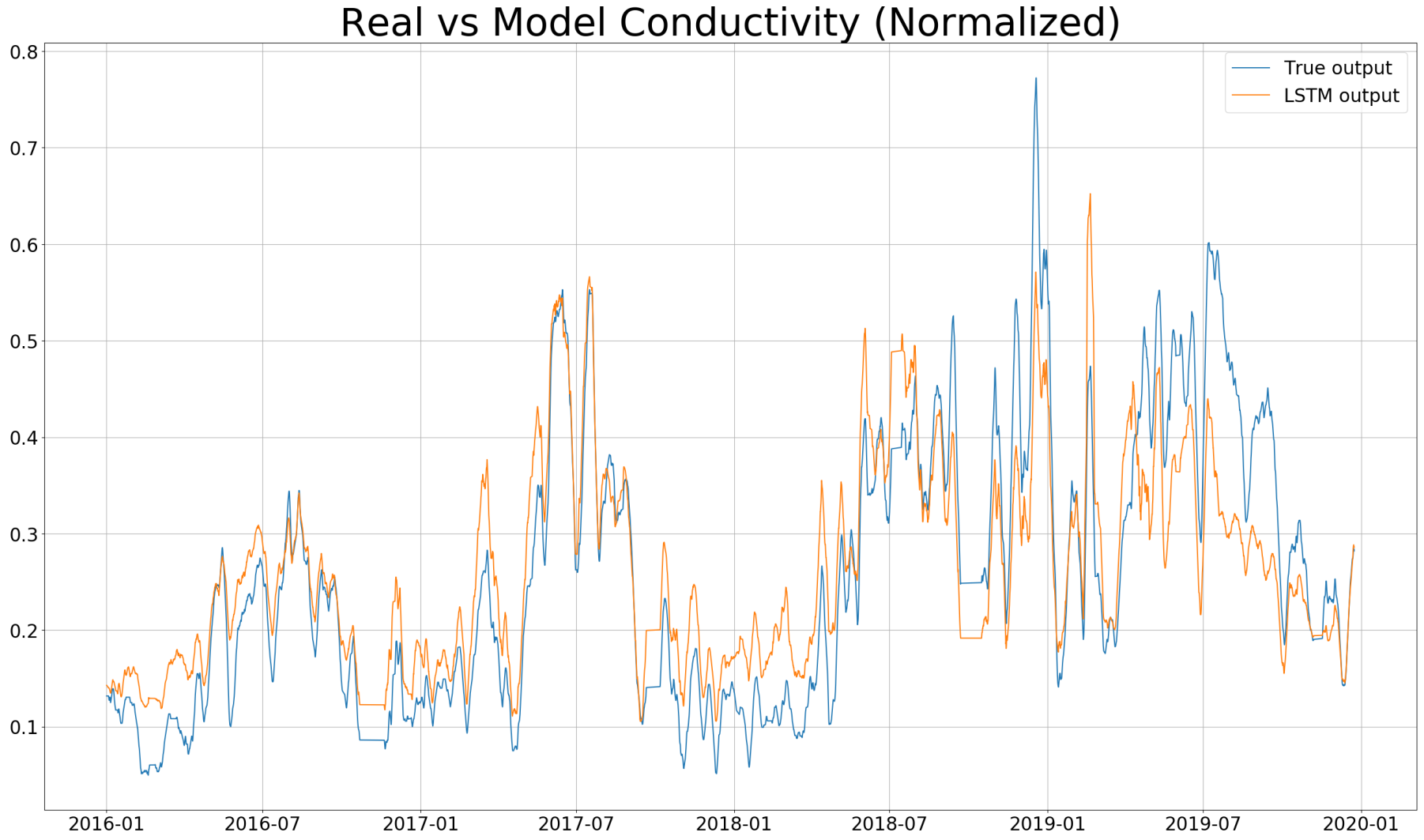

Figure B.32: Conductivity output of LSTM 6, compared to the true output.

B.9 LSTM 7

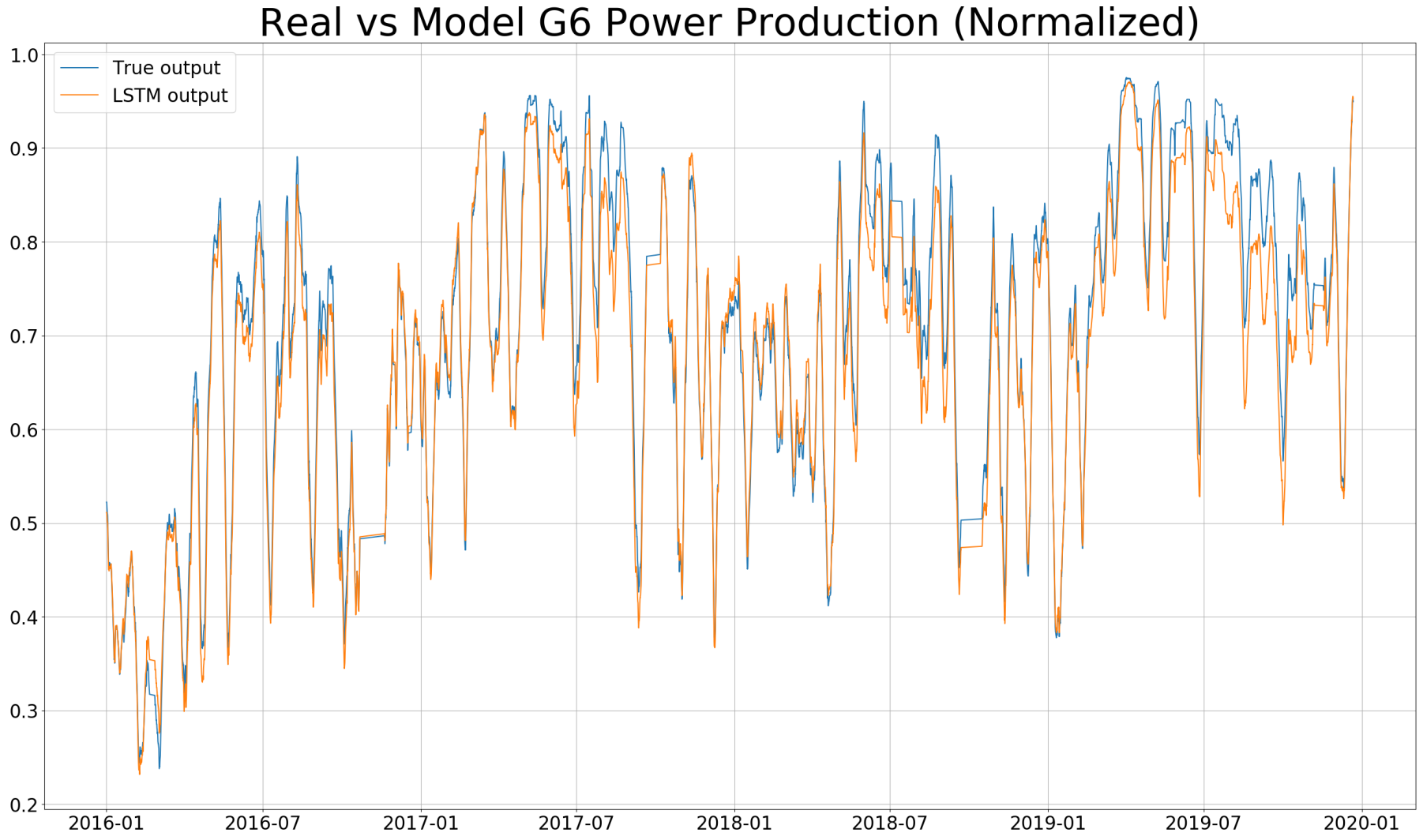

Figure B.33: Power production output of LSTM 7, compared to the true output.

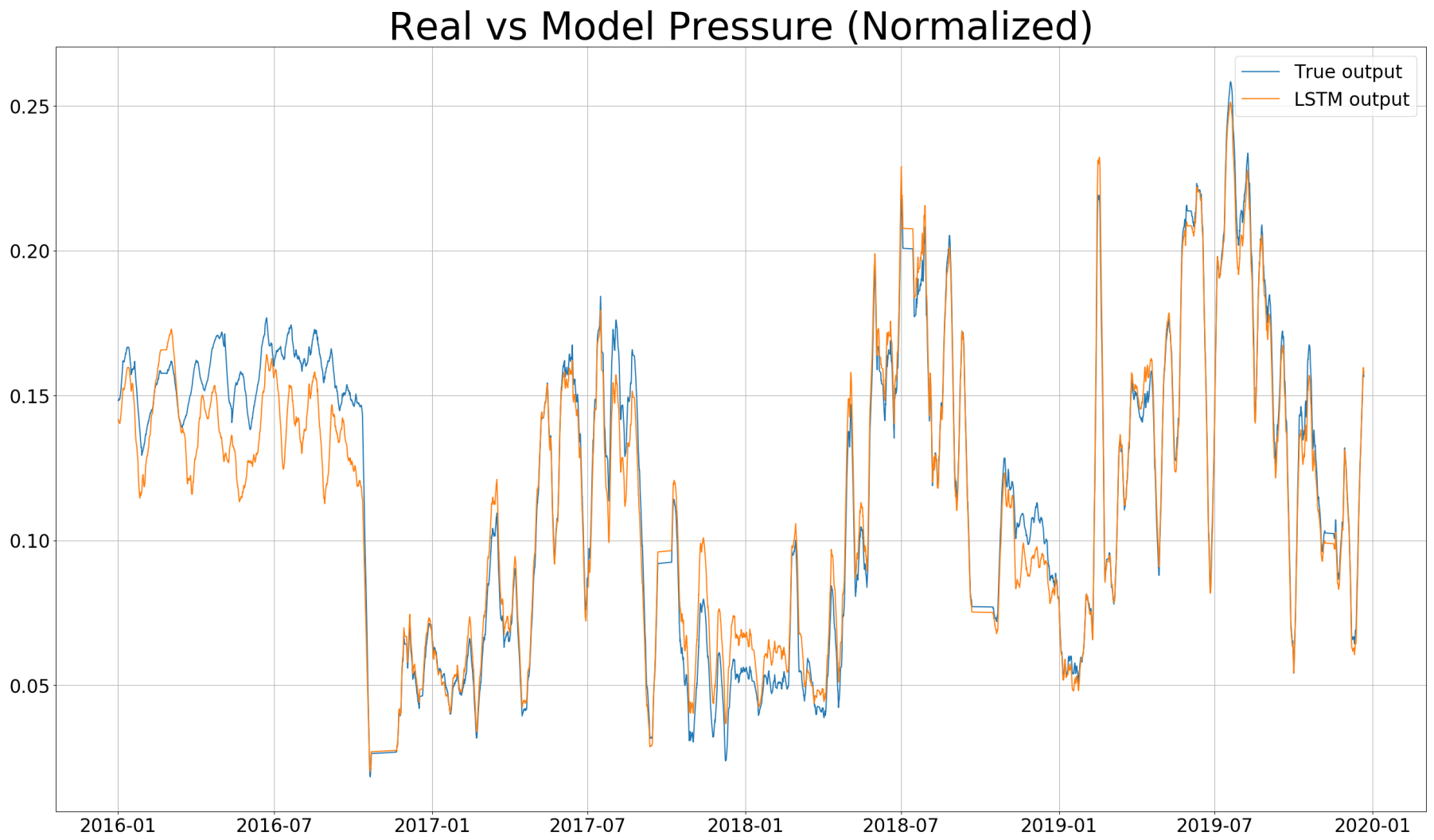

Figure B.34: Pressure output of LSTM 7, compared to the true output.

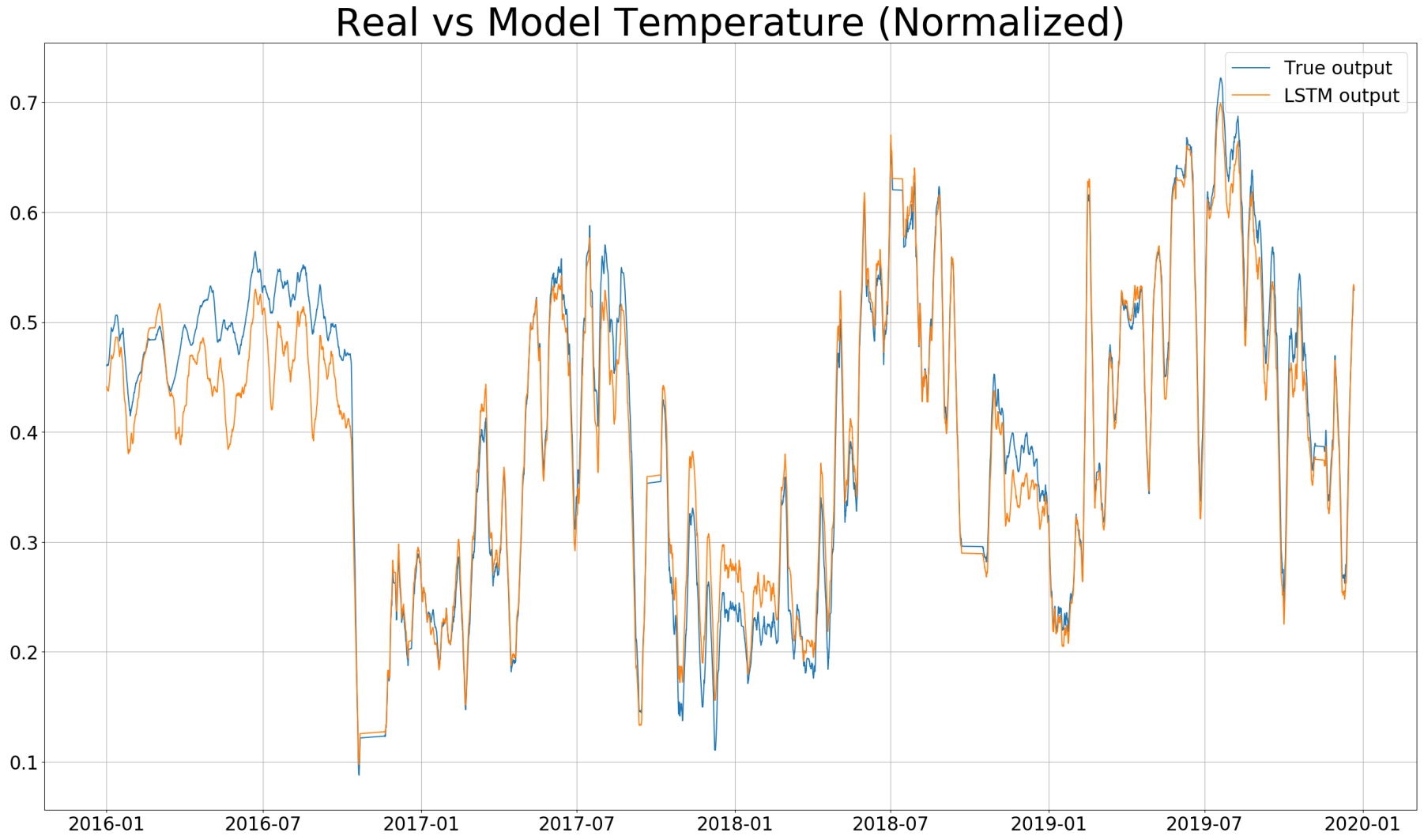

Figure B.35: Temperature output of LSTM 7, compared to the true output.

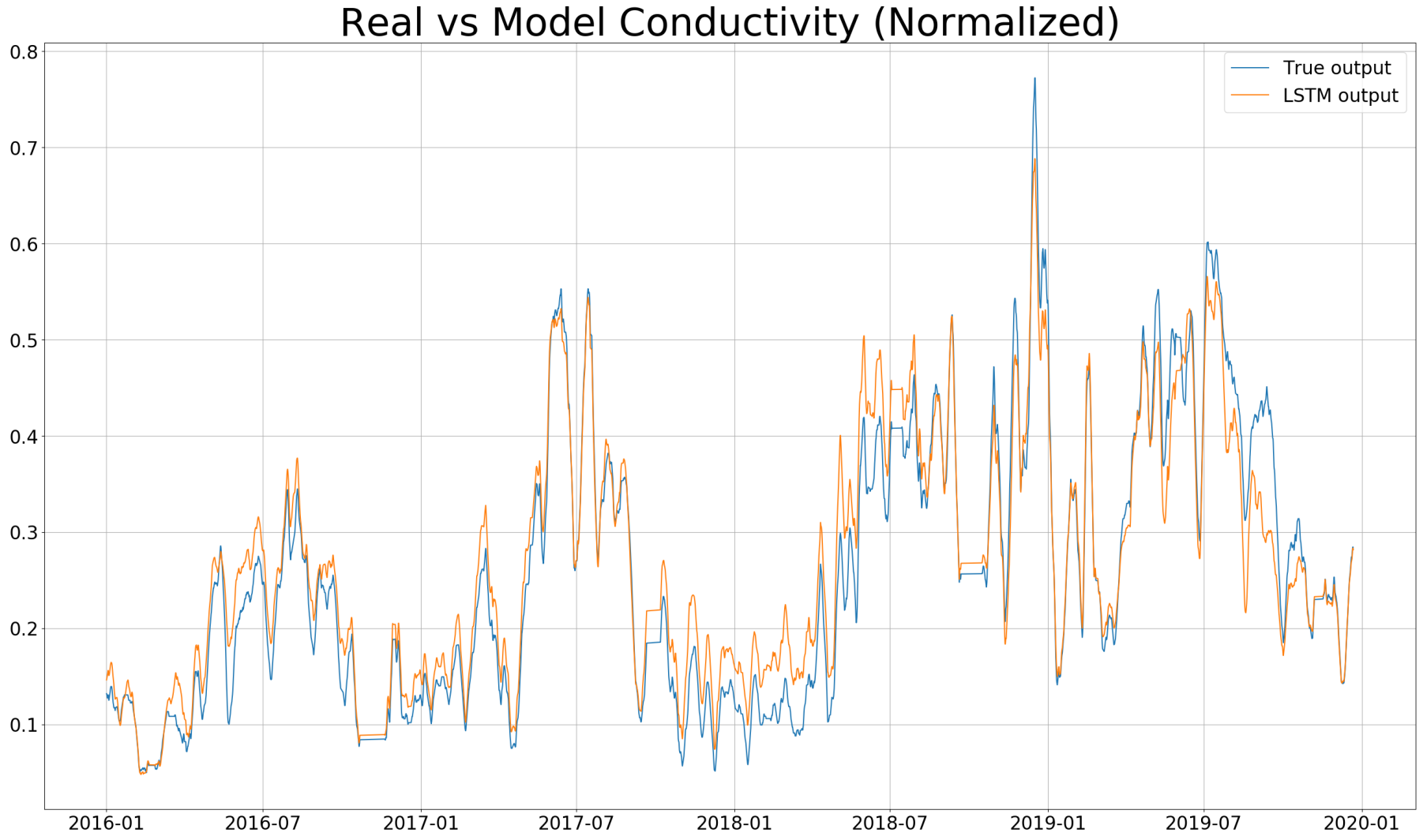

Figure B.36: Conductivity output of LSTM 7, compared to the true output.

B.10 LSTM 8

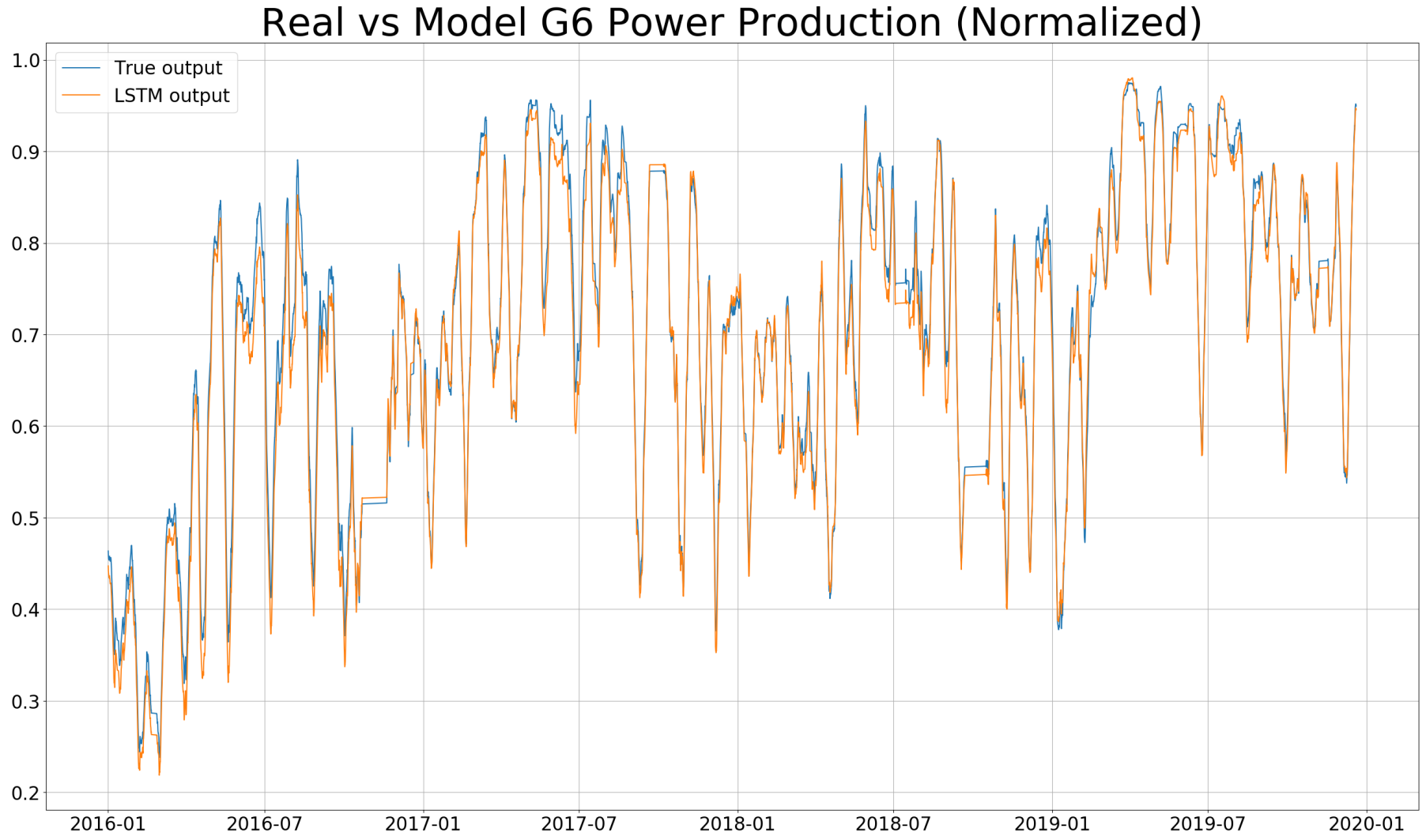

Figure B.37: Power production output of LSTM 8, compared to the true output.

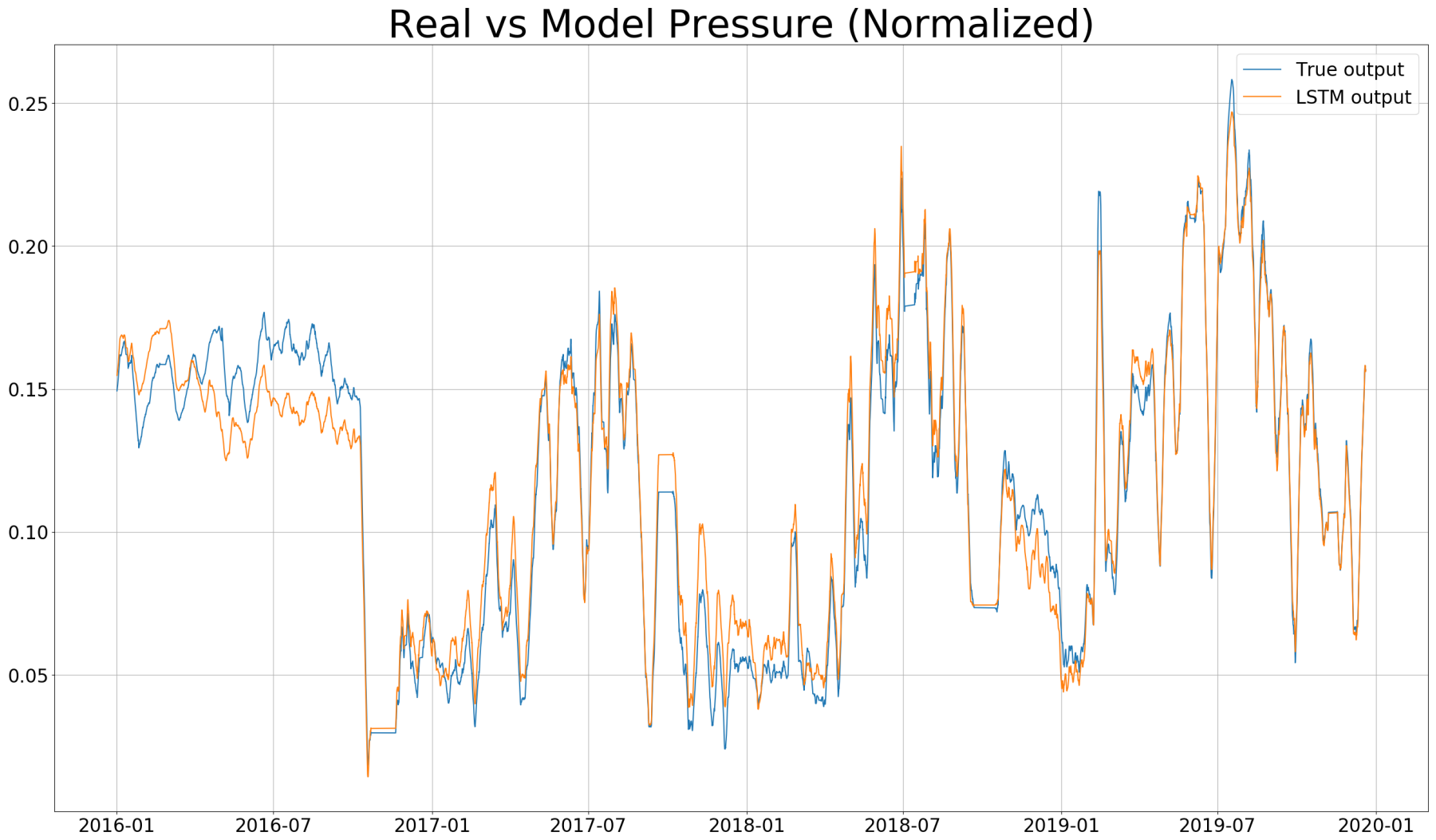

Figure B.38: Pressure output of LSTM 8, compared to the true output.

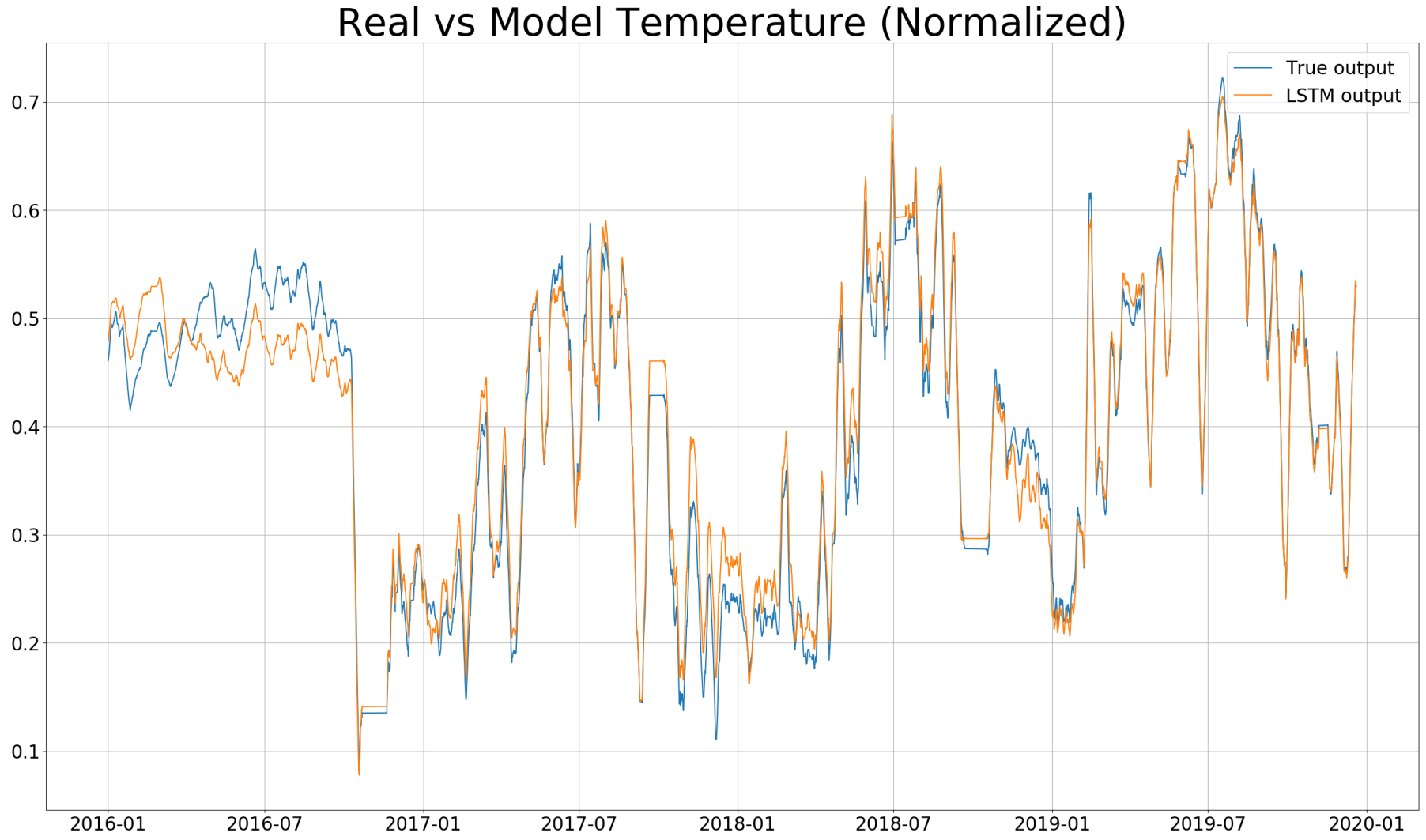

Figure B.39: Temperature output of LSTM 8, compared to the true output.

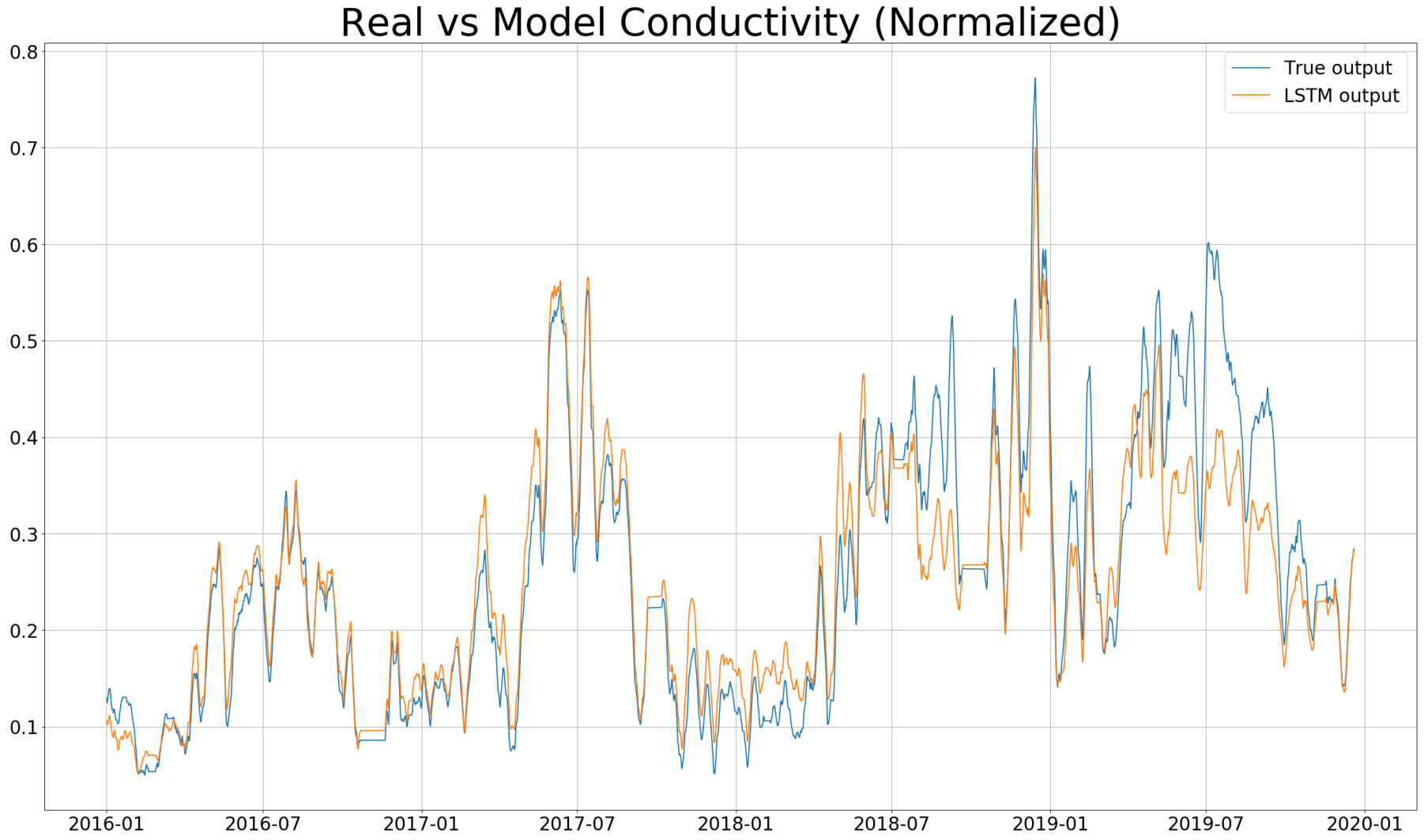

Figure B.40: Conductivity output of LSTM 8, compared to the true output.