IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 7, 2021, accepted October 19, 2021, date of publication October 21, 2021, date of current version November 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3121988

A Temporal Neural Network Model for Probabilistic Multi-Period Forecasting of Distributed Energy Resources

MARKUS LÖSCHENBRAND
SINTEF Energy Research, 7034 Trondheim, Norway
e-mail: markus.loschenbrand@sintef.no

This work was supported in part by the Centre for Intelligent Electricity Distribution (CINELDI), an Eight Year Research Centre under the FME-Scheme, Centre for Environment-Friendly Energy Research, under Grant 257626/E20; in part by the Research Council of Norway; and in part by the CINELDI Partners.

ABSTRACT Probabilistic forecasts of electrical loads and photovoltaic generation provide a family of methods able to incorporate uncertainty estimations in predictions. This paper aims to extend the literature on these methods by proposing a novel deep-learning model based on a mixture of convolutional neural networks, transformer models and dynamic Bayesian networks. Further, the paper also illustrates how to utilize Stochastic Variational Inference for training output distributions that allow time series sampling, a possibility not given for most state-of-the-art methods which do not use distributions. On top of this, the model also proposes an encoder-decoder topology that uses matrix transposes in order to both train on the sequential and the feature dimension. The performance of the work is illustrated on both load and generation time series obtained from a site representative of distributed energy resources in Norway and compared to state-of-the-art methods such as long-short-term memory. With a single-minute prediction resolution and a single-second computation time for an update with a batch size of 100 and a horizon of 24 hours, the model promises performance capable of real-time application. In summary, this paper provides a novel model that allows generating future scenarios for time series of distributed energy resources in real-time, which can be used to generate profiles for control problems under uncertainty.

INDEX TERMS Deep learning, generation forecasting, load forecasting, neural networks, probabilistic methods, renewable power.

NOMENCLATURE

Index

t	periods
B	batch

Functions

f	generic function notation
relu	rectified linear unit activation function
dense	fully connected linear layer
concate- nate	concatenation (i.e. 'stacking') of tensors
tanh	tangential hyperbolic activation function
σ	sigmoid activation function
softmax	normalized exponential function
softplus	softplus activation function
F	one-dimensional dilated convolutional kernel
KL	Kullback Leibler divergence
ELBO	Evidence LOwer Bound

Parameters

θ^{W}	linear weights
θ^{b}	linear biases
$\bar{\theta}^{W}$	linear weights (multiple channels)
$\bar{\theta}^{b}$	linear biases (multiple channels)
$\bar{\theta}^{res}$	convolution weights (residual branch)
$\bar{\theta}^{fil}$	convolution weights (filter branch)
$\bar{\theta}^{gate}$	convolution weights (gate branch)
a	series coefficients

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaodong Liang.

Variables

Y'	historical time series
X'	exogeneous variables of historical time series
y'	time series sample
x'	sampled exogeneous variables
v	generic tensor notation
s	generic matrix sequence notation
z_1'	sequential encoding
z_2	encoding of period $t+1$
g	distribution parameters of period $t+1$

Distributions

p	generic distribution notation
$\mathcal{N}$	Gaussian distribution
$\mathcal{B}$	Bernoulli distribution

I. INTRODUCTION

Increasing shares of renewable energy in the global mix of power generation lead to changes in the landscape of methods required to analyze and predict them. Whereas a classical, fossile fuel based power system behaves more static and gives more control over output levels to the power producer, more renewable generation means flexibility and thus more uncertainty. Such uncertainty is also amplified by another ongoing transition in power systems: increasing decentralization decreasing forecasting accuracy. This comes as individual loads or generation profiles of e.g. specific households or solar panels are harder to forecast than aggregates of several sources over larger areas [1], a result of higher variation on the individual level [2]. The result of these changes is a conceivable shift in the methods dealing with prediction of distributed energy resources from traditionally deterministic methods to methods incorporating uncertainty [3].

Methods accurately describing uncertainty are therefore at the center of the operational problems, be it optimization of storage under solar generation or utilization of shift-able loads as flexible assets [4], [5]. However, distributed energy resources do not only add to the growing importance of uncertainty, they also affect the time frames under which the models operate. Centralized energy systems offer large ranges of flexibility in production (e.g. in the form of thermal or hydropower plants), whereas the margins of such are smaller for distributed resources, thus also reducing the time horizons of the control problems [6]. An example is that of electrical storage: for large-scale, i.e. hydropower, the storage cycles range from days to months or years [7], where for distributed storage in form of batteries the operational cycles typically lie within single days [8]. This means that where large-scale storage allows for 'over-night' calculations of the operational decisions as well as the associated predictions [9], distributed energy resources are more sensible to computational times as they have to be applied in real-time [10].

Apart from a push towards incorporating uncertainty and real-time applications of forecasting of distributed energy resources, another trend is that of using highly non-linear over linearized prediction models. Amplified by recent achievements in machine learning, deep learning based methodologies have prevailed amongst these non-linear methods [11], [12]. This trend can also be attributed to new specialized hardware that allows scalable and parallel 'training' (i.e. finding the optimal parameters) of such models via batches of data. Amongst those deep-learning models, recurrent neural networks, specifically long-short-term memory neural networks have established themselves as the recent standard in load prediction, both for short-term [13]–[15] as well as long-term [16] applications. In addition to deterministic point-forecasts, these models have also been applied probabilistically [17], [18]. Another example of such is provided by [19] which shows how to train an autoregressive, non-linear quantile regressor based on long-short-term memory in order to provide probabilistic load flows. [20] provides a quantile regression example from PV prediction that uses a Lasso regressor. However, the output of such quantile regression methods is only represented via - as suggested by the name - quantiles. Thus, albeit probabilistic, the outputs of quantile regression do not provide the possibility to have individual samples drawn from as they only provide ranges and not distributions to sample from [21]. This is a problem that has been approached by [22], which presents a method to train deep learning models for load forecasting in form of distributions. However, in turn this presented model is an implementation of an auto-regressive neural network and thus does not incorporate temporality (i.e. the effects of state transitions over the time periods) with similar quality to the recurrent neural networks. With this current state of literature, readers are thus forced to choose between accurate representation of the mean or the ability to take samples from the distribution. In recent work, [23] approaches this issue via Bayesian regression, which in the here presented work is extended by replacing the non-linear decision tree approximator with deep neural networks, similar to the generative model presented in [22] but extended to auto-regression.

This is implemented via Stochastic Variational Inference [24], a method that can be utilized to train any parameterized distribution (in the here presented case a Gaussian and Bernoulli mixture model) via back-propagation, a technique that has been previously been applied in the domain of power systems to fit distributions in probabilistic optimal power flows [25] or make predictions in outlier events with data sets of small sample size [26].

To add to this contribution, the paper also analyzes the potential of convolutional neural networks as a replacement for recurrent neural networks in load forecasting. A similar model has been discussed in [27], however again with quantile losses and not considering temporality (the connection between periods in the input sequences), which recurrent neural networks do. Compared to this paper using traditional convolutional kernels, however, dilated convolutional neural networks have proven their capabilities in temporality, specifically in language prediction [28] and similarly applied in deterministic price forecasting within the power system [29].

Thus, the proposed model will build on these dilated convolutional kernels.

In recent literature on probabilistic load and generation forecasting tasks convolutional neural networks have been demonstrated to outperform recurrent neural networks. As such, [30] shows single-period forecast results for residential sites that match the analysis of convolutional neural networks provided in the multi-period case study below. Similar is done in [31]. Another example is provided by [32] where the authors propose a combination of recurrent and convolutional neural networks for the task of photovoltaic forecasting. Similar performance has been shown for quantile regression problems utilizing convolutional neural networks in [33] and [34].

In addition to an in-depth analysis on the state of the art of convolutional neural networks, this paper also illustrates how to incorporate attention mechanisms in such deep learning models. These attention mechanisms have been previously applied within the mentioned recurrent neural networks, whereas [35], [36] provide examples of such on the topic of load forecasting.

To incorporate this mechanism into temporal convolutional neural networks, the model presented in this paper will utilize a novel neural network layer, here titled a 'temporal convolutional attention' layer, which uses a similar topology to [37] mixed with the model from [28]. A similar idea has been previously proposed (deterministically and without the specific application on load forecasting) in [35], but the here presented model simplifies this attention mechanism to be more akin to the self-attention mechanism used in recurrent neural networks, thus reducing the amount of linear layers required to a third.

On top of these contributions, the paper also deals with an issue from the practical side of load forecasting - missing data. It does so by proposing an encoder-decoder architecture that encodes exogeneous variables within every time period and uses matrix transposes in order to propagate both over the dimension of variable and the dimension of sequence at the same time.

In summary, the contributions of this paper are to:

1) give an introduction of Stochastic Variational Inference as a method to train probabilistic load forecasts that allow taking individual samples from.
2) present a novel model that is a mixture between traditional convolutional neural networks, transformer models as presented in [37] and dynamic Bayesian networks. For the sake of simplicity[1] the model is from here on referred to as an Attention-based Temporal Convolutional Neural Network.
3) propose an encoder-decoder structure which uses matrix transposes to filter over sequential and feature dimensions in order to incorporate exogeneous variables into this network and more efficiently deal with missing input data.

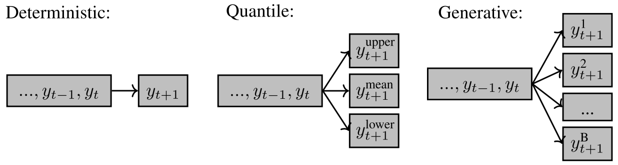


FIGURE 1. Comparison of auto-regression models.

In addition, the paper illustrates how to generate probabilistic load forecasts for multiple periods in advance and demonstrates this by predicting three heterogeneous time series obtained from a demonstration site representative of the Norwegian power grid.

By these contributions, the paper not only concludes in showing that Temporal Convolutional Neural Networks outperform other state-of-the-art techniques in probabilistic forecasting, it does so by providing a new state-of-the-art topology itself. Applications of the model are various, examples of which are solving load scheduling and coordination in distributed energy resources [4], [5], both competitive [18] or cooperative multi-agent problems [38] as well as stochastic energy management systems [39], [40].

Further, the model might also be combined with other models from literature into ensemble models [41] and/or even applied on a more granular level within individual buildings or on individual assets [42].

II. METHOD

Using the notation of $'$ to mark sequential data and assuming inputs consisting of historical values of the data series denoted as vector $Y' = [\ldots, y_{t-1}, y_t]$ and exogeneous variables denoted as matrix $X' = [\ldots, x_{t-1}, x_t]$, the ARX (autoregressive model with exogeneous variables) predicting the next future value can be defined similar to [43]:

$$y_{t+1} = f_\theta(X', Y') \tag{1}$$

In comparison to this deterministic problem, the probabilistic equivalent aims to fit a parameterized distribution p instead of a function f:

$$y_{t+1} \sim p_\theta(y_{t+1}|X', Y') \tag{2}$$

As such, a probabilistic model is not only making a single point prediction, but defining a range of outcomes. A comparison of the proposed probabilistic generative model to the deterministic and the probabilistic quantile regression models is provided in Fig. 1.

Therefore, the probabilistic regression problem is to first find an accurate model for approximating this distribution p and then apply a method that finds the best numerical fit for the parameters θ of the given model. These two problems will be approached successively, starting with presenting the proposed deep learning model first (forward pass) and then presenting the algorithm utilized to find the model parameters afterwards (backward pass). Contribution 3 is introduced in

[1] and to avoid confusion with electrical engineering terminology caused by phrases such as 'transformers'.

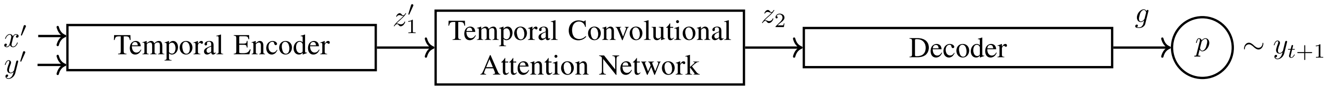


FIGURE 2. Proposed model structure.

the forward pass section, contribution 1 in the backward pass section and contribution 2 is demonstrated later together with the results in the case study section.

As all available input data Y' and its corresponding exogeneous variables X' will here be assumed too large for the model to consider as a whole, the model instead will utilize sampled batches from the given data sets. Sampling can be conducted by choosing a batch length B, a sample length τ and randomly sampling a number of time periods $t_1, \ldots, t_b$:

$$y' = \begin{bmatrix} [[y_{t_1-\tau}], \ldots, [y_{t_1-1}], [y_{t_1}]], \\ [[y_{t_2-\tau}], \ldots, [y_{t_2-1}], [y_{t_2}]], \\ \cdots \\ [[y_{t_B-\tau}], \ldots, [y_{t_B-1}], [y_{t_B}]] \end{bmatrix}$$
$$x' = \begin{bmatrix} [x_{t_1-\tau}, \ldots, x_{t_1-1}, x_{t_1}], \\ [x_{t_2-\tau}, \ldots, x_{t_2-1}, x_{t_2}], \\ \cdots \\ [x_{t_B-\tau}, \ldots, x_{t_B-1}, x_{t_B}] \end{bmatrix} \tag{3}$$

It has to be noted that the dimension of y' is increased by one here. The reason for this is to fit the input dimension of this batch of vectors with the batch of matrixes that is x' (with the most inner vector x_t presenting the features). This is discussed in the overview of the tensor shapes in the Appendix.

1) FORWARD PASS

The schematic for the proposed deep learning model consists of three main parts and is presented in Fig. 2.

These parts fulfill the following functions:

Temporal Encoder - this network encodes the temporal information provided by the exogeneous variables (i.e. the feature dimension of x'). This not only deals with missing data[2] but also incorporates long-term periodicity provided by the exogeneous variables into the input to the temporal convolutional network.

Temporal Convolutional Attention Network - this network is used to identify the patterns within the sequences of its input z' provided to it.

Decoder - this network generates the parameters utilized in the distribution to be sampled from.

As it can be observed from this illustration, the temporal encoding network and the output network are feed-forward neural networks mostly consisting of fully connected linear layers wrapped in 'rectified linear unit' activation functions:

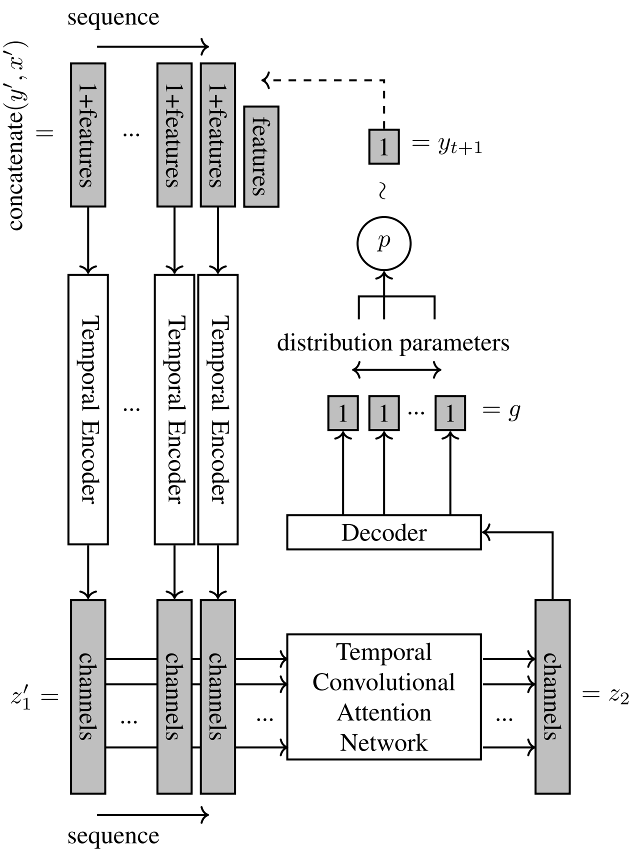


FIGURE 3. Forward flow (batch dimension for tensors omitted).

$$v^{\text{out}} = \text{relu}(\text{dense}_\theta(v^{\text{in}})) = \text{relu}(\theta^W v^{\text{in}} + \theta^b) \tag{4}$$

The only differences are that in the temporal encoder, the inputs x' and y' are concatenated (i.e. 'stacked') along the feature dimension. Further, the last layer of the decoder consists of several parallel layers (one for each distribution parameter).

The concatenation operation, together with the neural network applied on the feature dimension, allows to encode the information of the exogeneous variables (such as time stamps) into every single sequential step. This was done as inputs to traditional sequential networks (such as the here utilized temporal convolutional attention network or recurrent neural networks) have difficulties coping with missing values. In traditional auto-regressive networks utilizing only y' as input, a longer sequence of missing values would distort the input. By encoding the historical values y' with the exogeneous variables x', the missing values can instead be omitted.

[2] In traditional temporal convolutional neural networks, the missing data would have to be replaced with zeros. This however can lead to issues in case of too many missing values, which is the case in the provided case study. With this temporal encoder, however, missing values can be omitted as the position of a given y_t value in the input sequence y' is encoded via the corresponding exogeneous variables x_t.

Compared to the temporal encoder, the temporal convolutional attention network operates in the sequential dimension. A more intuitive illustration is presented in Fig. 3 which describes the entire forward flow of the neural networks. This also illustrates why the encoder and decoder are formulated as dense layers. The reason is that the encoder resembles a traditional regression problem and thus does not consider sequentiality, whereas the decoder resembles a mapping to the parameters of the output distribution and thus also operates in the feature dimension instead of the sequence dimension.

The utilized temporal convolutional attention network is derived from an extension to the 'wavenet' model presented in [28]. A similar network has previously been implemented underutilization of attention as shown in [35]. However, this formulation utilizes three dense layers (referred to as 'key'/'query'/'value') using the topology proposed in [37]. Here, instead, a single self-attention layer equivalent to the attention mechanism used traditionally in recurrent neural networks and as originally presented in [44] is applied, thus reducing the number of dense layers required to a third over the formulation presented in [35].

The temporal convolutional attention network is formulated as a series of temporal convolutional attention (tca) layers wrapped in rectified linear units:

$$s^{\text{out}} = \text{relu}(\text{tca}_\theta(s^{\text{in}})) \tag{5}$$

Using the notation from [28], i.e. $\odot$ to represent element-wise multiplication and $*$ to represent convolution, a single tca layer can be described the following:

$$\begin{aligned} \text{tca}(s^{\text{in}}) = {} & F_{\bar{\theta}^{res}} * \big(\text{softmax}(\bar{\theta}^W s^{\text{in}} + \bar{\theta}^b) \odot s^{\text{in}}\big) \\ & + \tanh\Big(F_{\bar{\theta}^{fil}} * \big(\text{softmax}(\bar{\theta}^W s^{\text{in}} + \bar{\theta}^b) \odot s^{\text{in}}\big)\Big) \\ & + \sigma\Big(F_{\bar{\theta}^{gate}} * \big(\text{softmax}(\bar{\theta}^W s^{\text{in}} + \bar{\theta}^b) \odot s^{\text{in}}\big)\Big) \end{aligned} \tag{6}$$

This is also graphically displayed in Fig. 4. Here it has to be noted that both the dense layers as well as the convolutional kernels have multiple channels (thus using the notation of $\bar{\theta}$), which, as also illustrated previously in Fig. 3, correspond exactly to the number of channels derived by the temporal encoder.

Using T to denote the matrix transpose, the forward flow of the proposed model can thus be formulated as shown in Algorithm 1.

For simplification and similar to Fig. 3, the batch dimensions have been omitted here. For the same reason, the weights θ have not been numbered. Nonetheless, every instance of θ represents an individual set of weights (or biases).

Further, the chosen distribution of $p(y_{t+1}|g)$ might vary depending on application and, similarly to other probabilistic models, has to be chosen dependent on the application. Usually, due to the central limit theorem, the most suitable distribution for an application with no further information on the real distribution can be assumed to be that of a Gaussian. This is also the case for the here proposed application of electrical load and generation forecasting. However, as also generation stemming from solar panels were analyzed, it was chosen to add a Bernoulli distribution to accurately model the day and night cycles. This additional information can be added via formulating the chosen distribution the following:

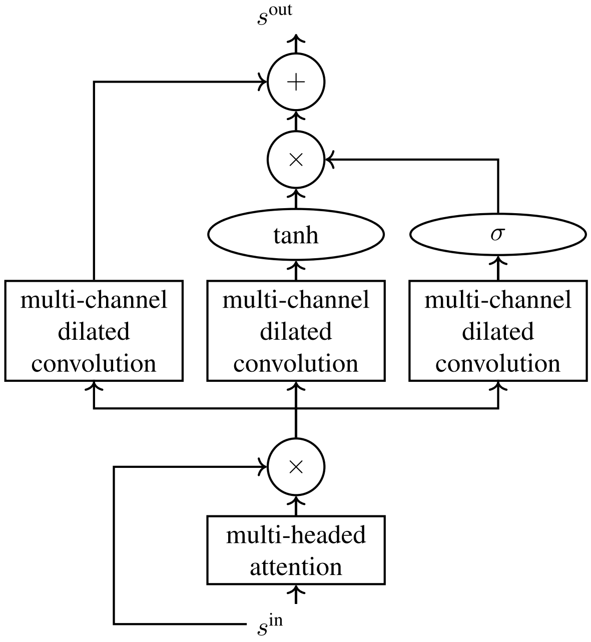

s^{out}
+
×
tanh
σ
multi-channel dilated convolution
multi-channel dilated convolution
multi-channel dilated convolution
×
multi-headed attention
s^{in}

FIGURE 4. Temporal convolutional attention (tca) layer.

Algorithm 1: Forward Flow of $y_{t+1} \sim p_\theta(y_{t+1}|x', y')$

```
sample: $y'$ and $x'$ from $Y'$ and $X'$;
$v' = \text{concatenate}(y', x')$ ;
for number of encoder layers $-1$ do
    $v' := \text{relu}(\text{dense}_\theta(v'))$ ;
$z'_1 = \text{dense}_\theta(v')$ ;
$s = z'^T_1$ ;
for $\sqrt{\tau} - 1$ convolutional attention layers do
    $s := \text{relu}(\text{tca}_\theta(s))$ ;
$z_2 = \text{tca}_\theta(s)$ ;
$v = z^T_2$ ;
for number of decoder layers $-1$ do
    $v := \text{relu}(\text{dense}_\theta(v))$ ;
$g = \bar{\theta}^W v^{\text{in}} + \bar{\theta}^b$ ;
sample: $y_{t+1} \sim p(y_{t+1}|g)$;
```

Algorithm 2: $p(y_{t+1}|g)$ Used in Case Study

```
sample: $y^{\mathcal{N}}_{t+1} \sim \mathcal{N}(g_1, \text{softplus}(g_2))$;
sample: $y^{\mathcal{B}}_{t+1} \sim \mathcal{B}(\sigma(g_3))$;
$y_{t+1} = y^{\mathcal{N}}_{t+1} \odot y^{\mathcal{B}}_{t+1}$ ;
```

Assuming there was no output distribution but instead deterministic outputs as in Eq. (1), these models could be trained via finding the mean squared error (or a similar loss measurement), backpropagation (i.e. finding the gradients) and applying an optimizer (i.e. a method such as stochastic gradient descent that updates the weights and biases). However, as the outputs are samples from a distribution instead a bounding function, namely the Evidence Lower BOund (short: ELBO), is applied here.

2) BACKWARD PASS

Assuming the measurement is the distance between the parameterized distribution p_θ from Eq. (2) and the real distribution p, the ELBO can be derived from the Kullback Leibler divergence (a distance measure for two distributions) as shown in [45]:

$$\begin{aligned}&\mathrm{KL}\big(p_\theta(y_{t+1}|v'), p(y_{t+1}|v')\big)\\&\quad= \int_{y_{t+1}} p_\theta(y_{t+1}|v') \log\big(\frac{p_\theta(y_{t+1}|v')}{p(y_{t+1}|v')}\big)\\&\quad= \int_{y_{t+1}} p_\theta(y_{t+1}|v') \log\big(\frac{p_\theta(y_{t+1}|v')}{p(v'|y_{t+1})p(v')}\big) + \log\big(p(v')\big)\\&\quad= -\mathrm{ELBO}(\theta) + \log\big(p(v')\big) \qquad (7)\end{aligned}$$

```
Algorithm 3: Backpropagation to Update $\theta$
    sample and fix: noise in Algorithm 2;
    conduct: forward flow from Algorithm 1;
    calculate: $\nabla ELBO(\theta)$;
    update: $\theta$;
```

```
Algorithm 4: Multi-Period Multi-Batch Prediction
    initialize: current time period as $t$
    for number of prediction periods do
        set: $y' = \begin{bmatrix}[[y_{t-\tau}], \dots, [y_{t-1}], [y_t]],\\ [[y_{t-\tau}], \dots, [y_{t-1}], [y_t]],\\ \dots\\ [[y_{t-\tau}], \dots, [y_{t-1}], [y_t]]\end{bmatrix}$;
             $x' = \begin{bmatrix}[x_{t-\tau}, \dots, x_{t-1}, x_t],\\ [x_{t-\tau}, \dots, x_{t-1}, x_t],\\ \dots\\ [x_{t-\tau}, \dots, x_{t-1}, x_t]\end{bmatrix}$
        sample: $y_{t+1} \sim p_\theta(y_{t+1}|x', y')$;
        set: t:= t+1;
```

Note that, for simplification, this equation considers a single input series $v' = \text{concatenate}(y', x')$. Due to $\log\big(p(v')\big)$ being a constant, an ELBO of 0 thus means a perfect fit for the Kullback Leibler divergence and thus a perfect fit of the output of the neural network based on the historical values y', x' on the distribution p_θ of the future data y_{t+1}.

[46] shows that for a model with a similar encoder-decoder architecture training can be accelerated by fixing the errors prior to obtaining the gradients of this ELBO function. Doing so, the process of Stochastic Variational Inference [24] for the given deep-learning model thus becomes the following step-wise updating process:

The optimization algorithm utilized for the weight update steps chosen in the following case study was a batch optimization algorithm that can be found in [47].

For training this model the process of updating the parameters is repeated for a given number of episodes (or until a satisfying ELBO is reached). After this, and similar to the model presented in [28], the network can be used over a rolling horizon to predict multiple periods in the future. This, as previously shown in Fig. 3 is realized via applying Algorithm 4.

The output of this process is a batch of sequences which represent samples of future expectations. In the following section this model is demonstrated by a case study of three heterogeneous time series and compared to other state-of-the-art models.

III. CASE STUDY

The endogeneous data Y for the given case study was obtained from a test location designed to represent timely issues in the Norwegian power grid. At the core of the application stands the modeling of battery energy storage systems and load shifts in HVAC systems under consideration of three distinctively heterogeneous time series, two consumption time series - a residential load (C1) and an office building connected to a football stadium (C2), as well as a time series of photovoltaic generation with a capacity of 800kW (PV). Even though commercially run, the test site is designed to accurately represent greater challenges in the Norwegian power grid, such being strong reactions to temperature changes, highly fluctuating weather conditions, seasonal as well as other patterns and stochastic events of increased consumption.

The available resolution of 1 minute from the sensors was kept, whereas the sensors where not operational continuously during the measurement period, leading to the previously discussed missing values with frequent outage sequences of up to 60 minutes. Fig. 5 gives a visual overview of two months of the data series (in its entirety 305 days long). The electrical demand of series C1 shows day and night consumption patterns with occasional low-amplitude outliers. Series C2 shows conceivable weekday/weekend patterns with high amplitude outliers due to operation of the stadium (mainly caused by floodlights) in addition to these cycles from C1. Series PV shows periodicity due to day/night differences and cloud patterns during the days. Consistent with the requirements expressed by the test site owners the prediction horizon was selected to be 1440 minutes, with the model being updated by the sensors in real-time, i.e. in 1 minute tacts.

The exogeneous variables X for the given case study were derived solely on these presented time series with no additional information such as weather or temperature supplied, a result of the lower resolution of the available

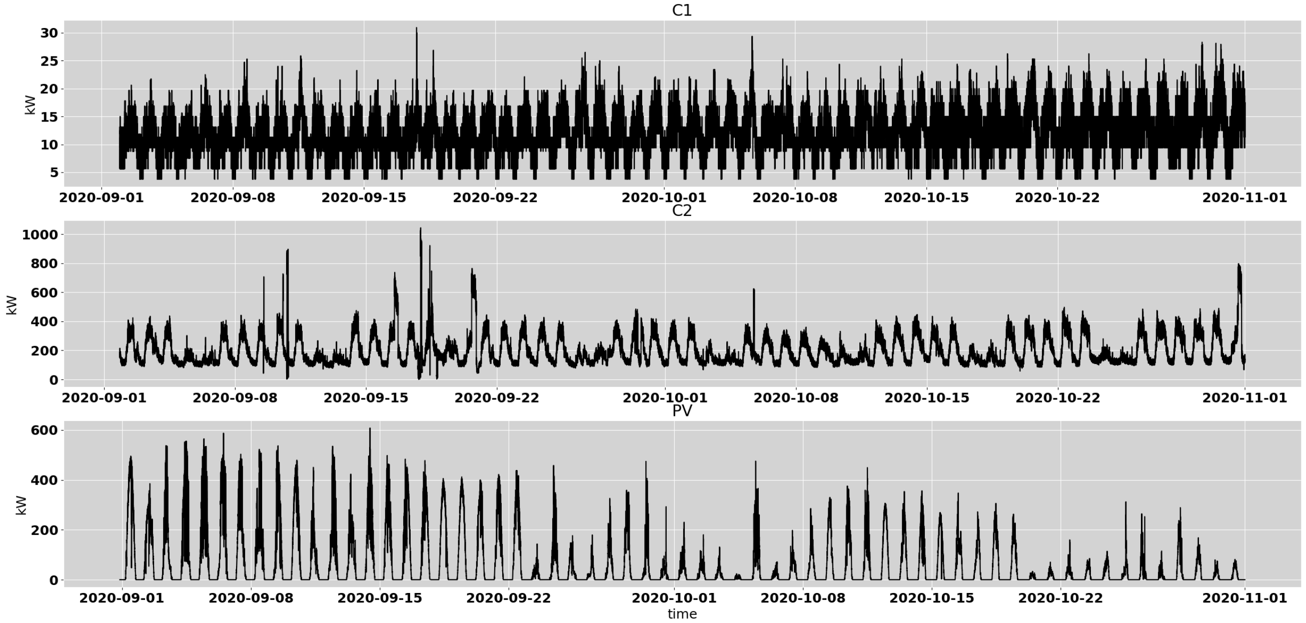
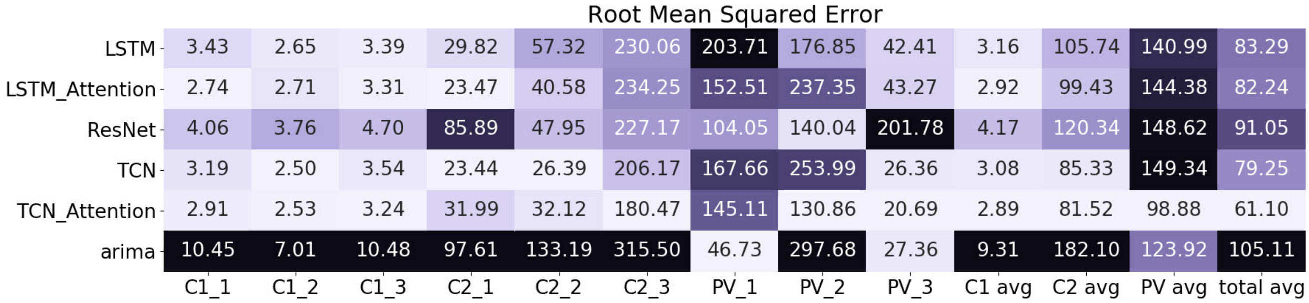


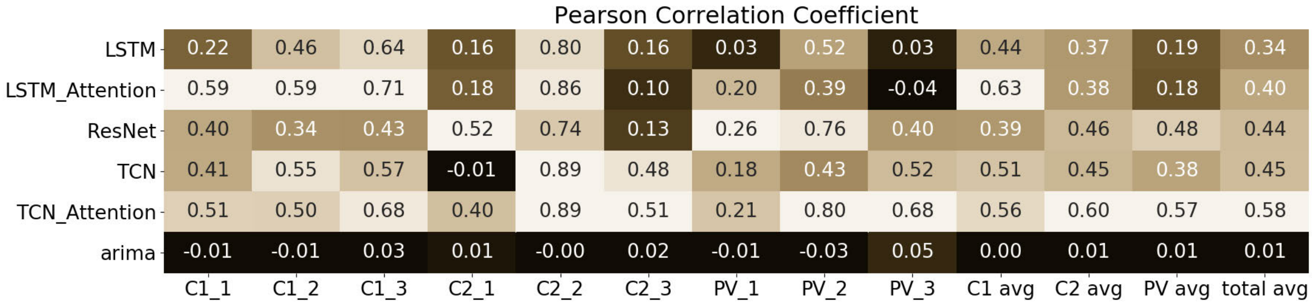
FIGURE 5. Case study: data set excerpt.



Root Mean Squared Error

	C1_1	C1_2	C1_3	C2_1	C2_2	C2_3	PV_1	PV_2	PV_3	C1 avg	C2 avg	PV avg	total avg
LSTM	3.43	2.65	3.39	29.82	57.32	230.06	203.71	176.85	42.41	3.16	105.74	140.99	83.29
LSTM_Attention	2.74	2.71	3.31	23.47	40.58	234.25	152.51	237.35	43.27	2.92	99.43	144.38	82.24
ResNet	4.06	3.76	4.70	85.89	47.95	227.17	104.05	140.04	201.78	4.17	120.34	148.62	91.05
TCN	3.19	2.50	3.54	23.44	26.39	206.17	167.66	253.99	26.36	3.08	85.33	149.34	79.25
TCN_Attention	2.91	2.53	3.24	31.99	32.12	180.47	145.11	130.86	20.69	2.89	81.52	98.88	61.10
arima	10.45	7.01	10.48	97.61	133.19	315.50	46.73	297.68	27.36	9.31	182.10	123.92	105.11

FIGURE 6. Average root mean squared error per minute.



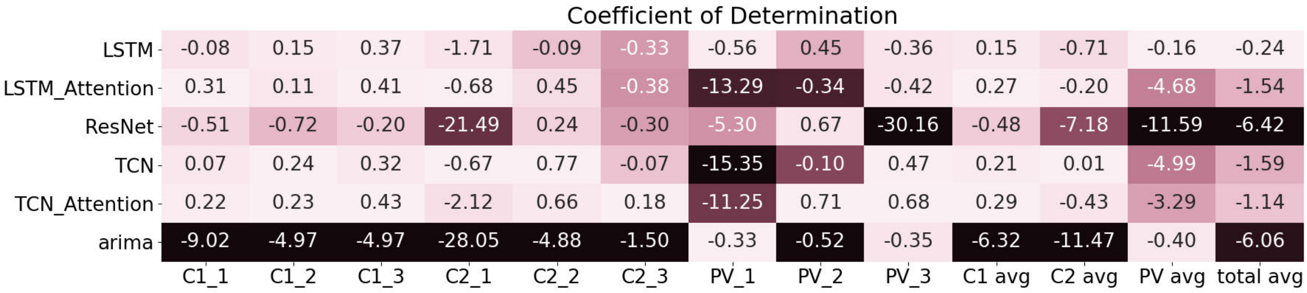
Pearson Correlation Coefficient

	C1_1	C1_2	C1_3	C2_1	C2_2	C2_3	PV_1	PV_2	PV_3	C1 avg	C2 avg	PV avg	total avg
LSTM	0.22	0.46	0.64	0.16	0.80	0.16	0.03	0.52	0.03	0.44	0.37	0.19	0.34
LSTM_Attention	0.59	0.59	0.71	0.18	0.86	0.10	0.20	0.39	-0.04	0.63	0.38	0.18	0.40
ResNet	0.40	0.34	0.43	0.52	0.74	0.13	0.26	0.76	0.40	0.39	0.46	0.48	0.44
TCN	0.41	0.55	0.57	-0.01	0.89	0.48	0.18	0.43	0.52	0.51	0.45	0.38	0.45
TCN_Attention	0.51	0.50	0.68	0.40	0.89	0.51	0.21	0.80	0.68	0.56	0.60	0.57	0.58
arima	-0.01	-0.01	0.03	0.01	-0.00	0.02	-0.01	-0.03	0.05	0.00	0.01	0.01	0.01

FIGURE 7. Average Pearson correlation coefficient per minute.

weather data over the 1 minute resolution of the electrical loads.. Specifically, the additional exogeneous variables were a linear trend line, the day of the week, a binary series separating between weekdays and weekends, the quarter of the year, the hour, the minute, the day of the year and the month. In addition, Fourier series were added as exogeneous variables representing periodicity:

$$\begin{aligned} X_a'^{\sin} &= \left[\sum \sin(\frac{\pi}{a}t) \quad \forall t\right] \\ X_a'^{\cos} &= \left[\sum \cos(\frac{\pi}{a}t) \quad \forall t\right] \end{aligned} \tag{8}$$



Coefficient of Determination

	C1_1	C1_2	C1_3	C2_1	C2_2	C2_3	PV_1	PV_2	PV_3	C1 avg	C2 avg	PV avg	total avg
LSTM	-0.08	0.15	0.37	-1.71	-0.09	-0.33	-0.56	0.45	-0.36	0.15	-0.71	-0.16	-0.24
LSTM_Attention	0.31	0.11	0.41	-0.68	0.45	-0.38	-13.29	-0.34	-0.42	0.27	-0.20	-4.68	-1.54
ResNet	-0.51	-0.72	-0.20	-21.49	0.24	-0.30	-5.30	0.67	-30.16	-0.48	-7.18	-11.59	-6.42
TCN	0.07	0.24	0.32	-0.67	0.77	-0.07	-15.35	-0.10	0.47	0.21	0.01	-4.99	-1.59
TCN_Attention	0.22	0.23	0.43	-2.12	0.66	0.18	-11.25	0.71	0.68	0.29	-0.43	-3.29	-1.14
arima	-9.02	-4.97	-4.97	-28.05	-4.88	-1.50	-0.33	-0.52	-0.35	-6.32	-11.47	-0.40	-6.06

FIGURE 8. Average "R-squared" metric.

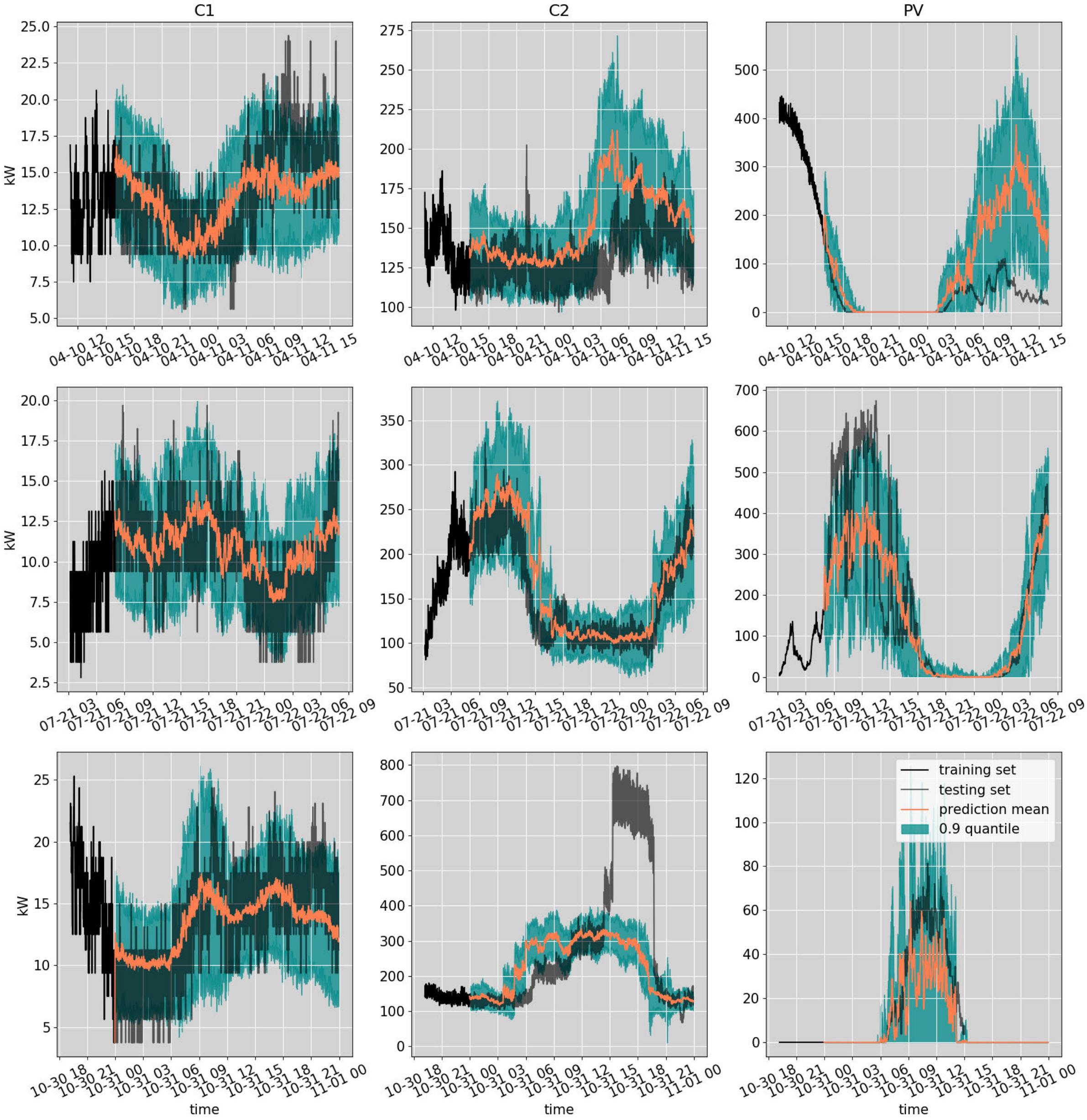


FIGURE 9. Visual results for the proposed model.

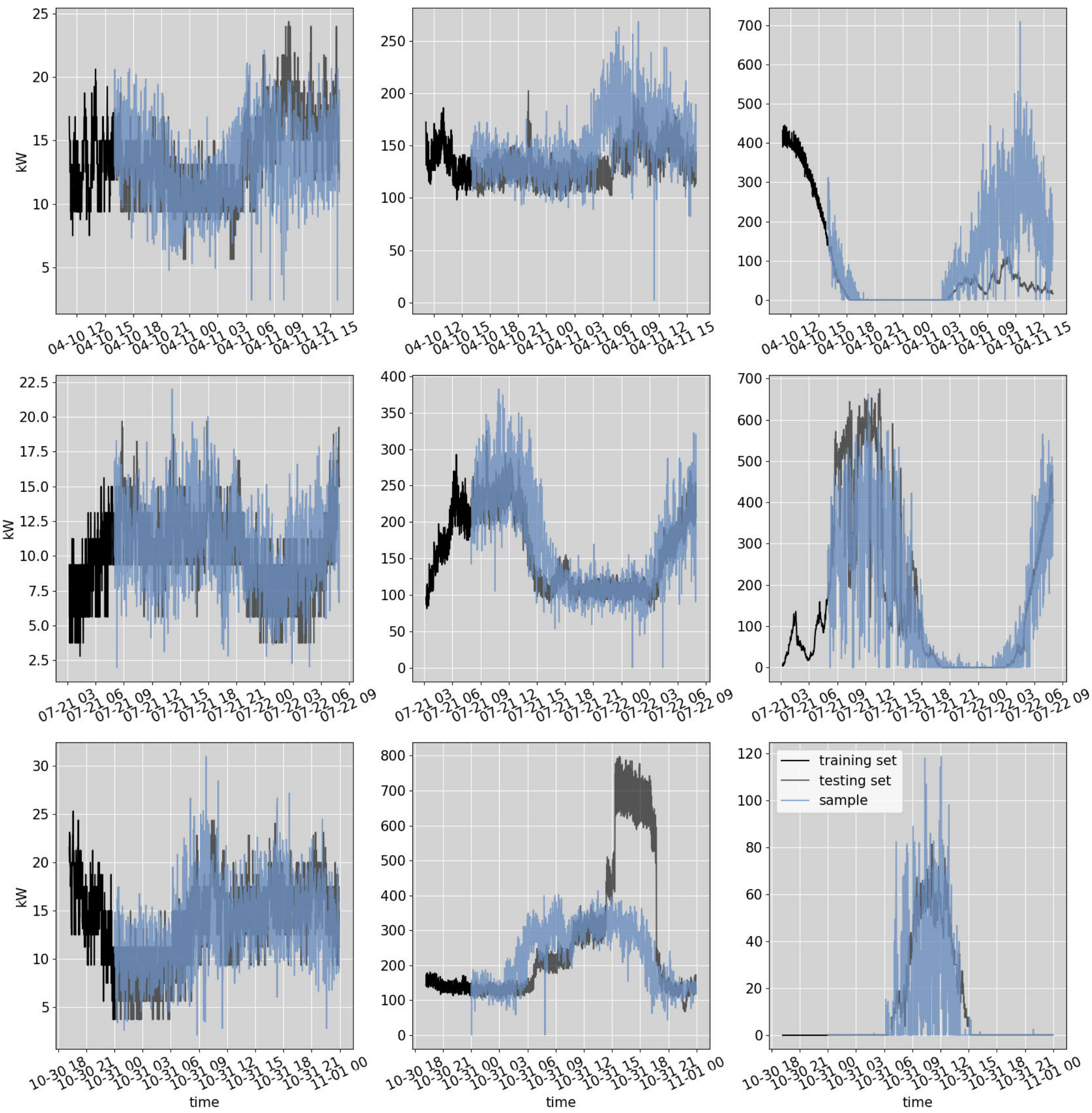


FIGURE 10. Single sample taken from proposed model.

The coefficients chosen for these series where 5, 10, 15, 30, 60, 120, 720 minutes ($a = 5, 10, 15, 30, 60, 120, 720$), a day ($a = 1440$) and a week ($a = 10080$).

In addition to this, both X' and Y' were standardized before feeding it to the model and batch normalization layers were added (behind all the linear layers, except the output layers) in order to support convergence of the model.

In the case study, the following models were compared:

ARIMA - an auto-regressive moving average model, as presented e.g. in [12],

ResNet- a residual neural network, as shown in [22], [48],[3]

LSTM - a long-short-term memory neural network, as shown in [13], [14], [16], [18], [50],

LSTM_attention - a long-short term memory neural network utilizing attention, as presented in [36], [51],

TCN - a temporal convolutional neural network, as presented in [35] and [30],[4]

TCN_attention - the model as proposed in this paper.

For the case study, each input data set was split into three equally sized chunks indicated by *_1, *_2, *_3, with the last day (=1440 periods) of each data set selected as the prediction target.

For the sake of comparison for all of these models, except for the ARIMA model the encoder and decoder section of

[3] An in-depth discussion on why generative adversarial neural networks as presented in [22] generalize to the encoder-decoder structure presented here is found in [49].

[4] Using the original wavenet block implementation from [28].

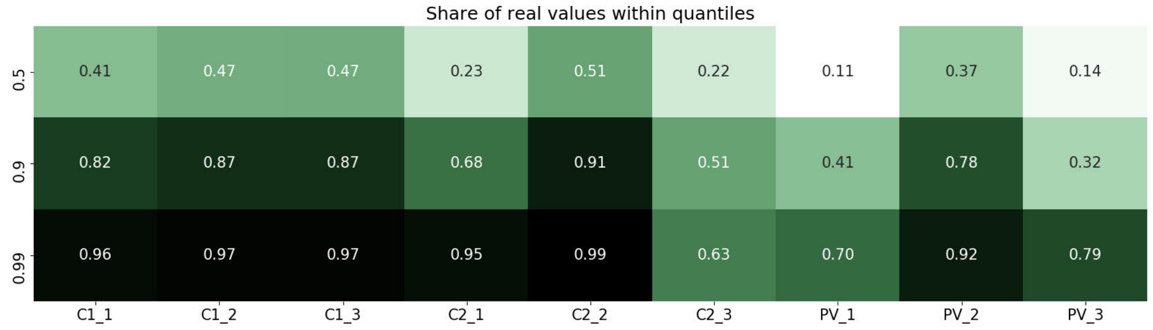


FIGURE 11. Quantile analysis.

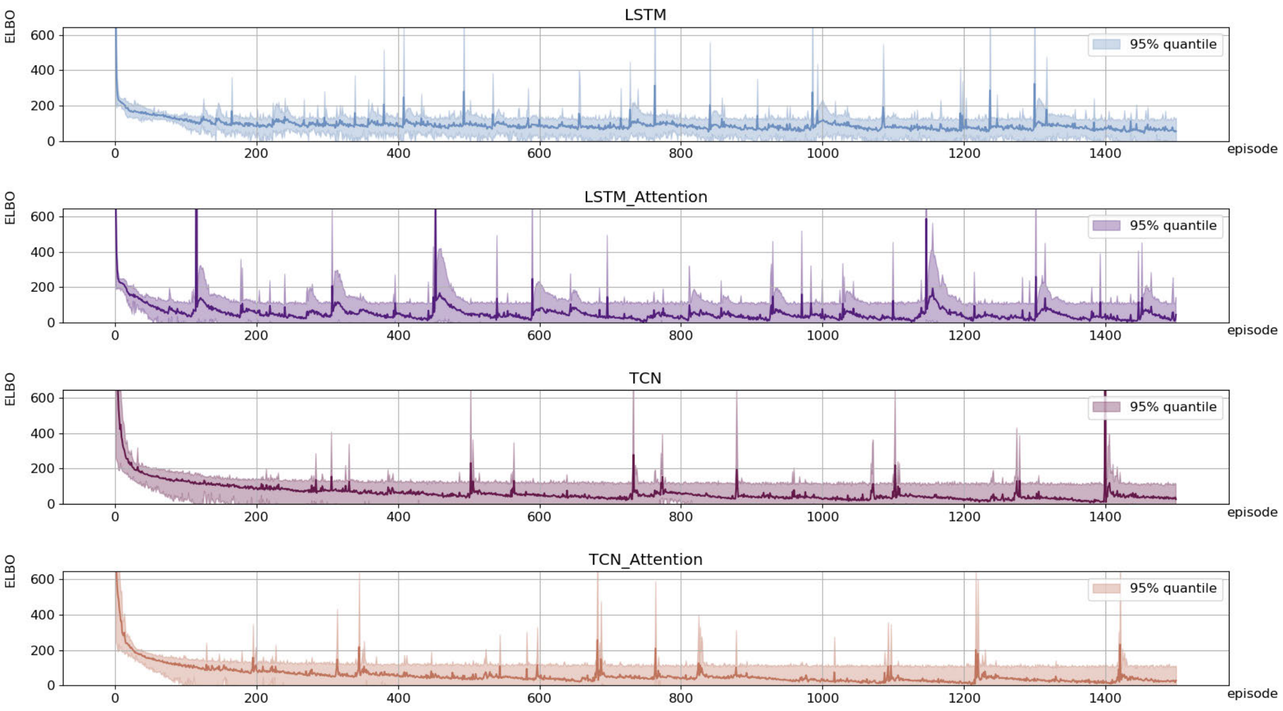


FIGURE 12. Training set loss curves.

the original model were kept intact and Stochastic Variational Inference was utilized to train all models. The encoder layer size (channels) was kept at 50, the linear decoder layer size (nodes) at 500. As the model is probabilistic, dropout layers were not considered to be required. Further, the given parameters were arbitrarily selected, under consideration of the number of features and available GPU capacity.

Each model was trained on a Nvidia Quadro P2000 and trained for 1500 episodes with a batch size of 100 and a training time of around 1 sec per episode for each model. An input time series length of five days was utilized (in order to capture weekend/weekday changes in the time series), but these input sequences were down-sampled to 100 periods each. For the channel size 50 layers was chosen (similar for the hidden size of the long-short-term memory models) and all linear layers were set to a size of 500 nodes. The AR and MA components of the ARIMA model were selected to be 60 periods each.

The root mean squared errors averaged over the 1440 test periods and all taken samples are given in Fig. 6. As it can be observed the proposed model performs either the best or amongst the best in all given data sets. Similar can be observed comparing the correlation coefficients between the samples and the real data points in Fig. 7. It has to be noted that these values are not calculated based on the mean of the samples but instead are calculated individually for each sample and then averaged. Further, the coefficient of determination as shown in Fig. 8 also supports the performance of the proposed model.[5]

[5] with the sole outlier of the series PV1, where the LSTM performs the best despite the highest RMSE on the similar series.

The capabilities of the model are also further outlined in the visual results provided by Fig. 9. As discussed before, as a generative model, this technique allows drawing single samples, an example of such is presented in Fig. 10. These samples come in form of time series and represent the distribution if drawn infinitely. Further note that the model indicates to capture variance as well as trends, both non-linear and periodical, even with a relatively small training set of 3.5 months. The main issues the model encounters in the test set are the following:

C2_1 - correctly anticipates the 'step', but does so too early.
C2_3 - does not anticipate the large outlier (this outlier is also presented in the last day shown in Fig. 5, which shows that it is in fact a rare occurrence).
PV_1 - overestimates the amplitude wrongly after 8:00am.
PV_3 - underestimates the amplitude of the peak around noon.

This synapsis is also supported by the quantiles shown in Fig. 11, which summarizes the number of the real outcomes found within the given quantiles which shows C2_1 to be the best performer of the outliers.

Nonetheless, and as shown in Figs. 6 and 7, the TCN_Attention model in general mostly outperforms the tested alternatives, even in the discussed outlier situations.

This is also supported by the training set losses as shown in Fig. 12 which shows generally more robustness to outliers in training on TCN.

Thus, in summary it can be stated that even though the model fails to capture unforeseen, rare events it still manages to accurately represent the underlying distribution of the time series better than the current state-of-the-art models. In addition to that, the results also indicate that even though the proposed attention mechanism improves the performance of the model significantly (and should thus be advised to be utilized), the temporal convolutional network still manages to compete with current state-of-the-art algorithms without this adjustment.

IV. CONCLUSION

This paper proposes a novel multi-period probabilistic load and generating forecasting model for distributed energy resources based on convolutional neural networks and a transformer-like stacked self-attention mechanism. Further, it also introduces Stochastic Variational Inference as a method to train probabilistic forecasting models that allows training any selected output distribution. As a generative method, it allows for taking samples of the output, a possibility not provided by other models based on mechanisms such as quantile regression. In addition to that the model also proposes an encoder-decoder structure in order to 'fill' the gaps of missing sensor data in the input.

The proposed model is then trained on chunks of data sets obtained from a site representative of the Norwegian power system - two consumer and one producer load series. The case study not only demonstrates the better performance of the proposed models compared to current state-of-the-art models but also highlights the performance of the temporal convolutional neural network being on-par with the state-of-the-art without applying the proposed attention mechanism.

In summary, this paper does not only introduce two principles that will aid future probabilistic load prediction - the encoder-decoder structure as well as the Stochastic Variational Inference for back-propagation - it also discusses the application of a novel neural network model on such tasks.

For future work, focus on more efficient generation of output sequences can be proposed, as well as larger case studies including weather and other exogeneous data can be suggested. As the outputs are generated auto-regressively, generating larger batches of samples can become time-inefficient for long prediction sequences.

In summary, it can be stated that the proposed probabilistic model provides an efficient method to generate samples of sequences for distributed energy resources in real-time, whose performance against several error measures is demonstrated in the context of representative datasets from the Norwegian power system.

APPENDIX
TENSOR SHAPES

Here, the dimensions of the tensors used in the model are listed. Please note that albeit x' is referred to as a matrix and y' is referred to as a vector, both of these variables are represented via tensors (due to the batch dimension):

y' - batch, sequence, 1
x' - batch, sequence, feature
z_1' - batch, sequence, channel
z_2 - batch, 1, channel
g - batch, distribution parameter, 1

ACKNOWLEDGMENT

The author would like to thank Johannes Philippus Maree, Venkatachalam Lakshmanan, and Iver Bakken Sperstad for the Inspiring Discussions on the Presented Method. Lede is gratefully acknowledged for the sharing of load data.

REFERENCES

[1] P. G. Da Silva, D. Ilic, and S. Karnouskos, "The impact of smart grid prosumer grouping on forecasting accuracy and its benefits for local electricity market trading," *IEEE Trans. Smart Grid*, vol. 5, no. 1, pp. 402–410, Jan. 2014.

[2] Y. Wang, Q. Chen, T. Hong, and C. Kang, "Review of smart meter data analytics: Applications, methodologies, and challenges," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 3125–3148, May 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8322199/

[3] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *Int. J. Forecasting*, vol. 32, no. 3, pp. 914–938, 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0169207015001508

[4] A. J. Conejo, M. Carrion, and J. M. Morales, *Decision Making Under Uncertainty in Electricity Markets*, vol. 1. New York, NY, USA: Springer, 2010.

[5] M. D. Somma, G. Graditi, E. Heydarian-Forushani, M. Shafie-Khah, and P. Siano, "Stochastic optimal scheduling of distributed energy resources with renewables considering economic and environmental aspects," *Renew. Energy*, vol. 116, pp. 272–287, Feb. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0960148117309382

[6] D. Brown, S. Hall, and M. E. Davis, "What is prosumerism for? Exploring the normative dimensions of decentralised energy transitions," *Energy Res. Social Sci.*, vol. 66, Aug. 2020, Art. no. 101475. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2214629620300529

[7] J. I. Pérez-Díaz, M. Chazarra, J. García-González, G. Cavazzini, and A. Stoppato, "Trends and challenges in the operation of pumped-storage hydropower plants," *Renew. Sustain. Energy Rev.*, vol. 44, pp. 767–784, Apr. 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1364032115000398

[8] J. Cao, D. Harrold, Z. Fan, T. Morstyn, D. Healey, and K. Li, "Deep reinforcement learning-based energy storage arbitrage with accurate lithium-ion battery degradation model," *IEEE Trans. Smart Grid*, vol. 11, no. 5, pp. 4513–4521, Sep. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9061038/

[9] O. Wolfgang, A. Haugstad, B. Mo, A. Gjelsvik, I. Wangensteen, and G. Doorman, "Hydro reservoir handling in Norway before and after deregulation," *Energy*, vol. 34, no. 10, pp. 1642–1651, Oct. 2009. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0360544209003119

[10] A. Goudarzi, A. G. Swanson, J. Van Coller, and P. Siano, "Smart real-time scheduling of generating units in an electricity market considering environmental aspects and physical constraints of generators," *Appl. Energy*, vol. 189, pp. 667–696, Mar. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0306261916318335

[11] H. Wang, "A review of deep learning for renewable energy forecasting," *Energy Convers. Manage.*, vol. 198, Oct. 2019, Art. no. 111799.

[12] X. Liu, Z. Zhang, and Z. Song, "A comparative study of the data-driven day-ahead hourly provincial load forecasting methods: From classical data mining to deep learning," *Renew. Sustain. Energy Rev.*, vol. 119, Mar. 2020, Art. no. 109632. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1364032119308391

[13] J. Zheng, C. Xu, Z. Zhang, and X. Li, "Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network," in *Proc. 51st Annu. Conf. Inf. Sci. Syst. (CISS)*, Baltimore, MD, USA, Mar. 2017, PP. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/7926112/

[14] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, "Short-term residential load forecasting based on LSTM recurrent neural network," *IEEE Trans. Smart Grid*, vol. 10, no. 1, pp. 841–851, Jan. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8039509/

[15] H. Zang, R. Xu, L. Cheng, T. Ding, L. Liu, Z. Wei, and G. Sun, "Residential load forecasting based on LSTM fusing self-attention mechanism with pooling," *Energy*, vol. 229, Aug. 2021, Art. no. 120682. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0360544221009312

[16] M. Dong and L. Grumbach, "A hybrid distribution feeder long-term load forecasting method based on sequence prediction," *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 470–482, Jan. 2020. [Online]. Available: https://ieeexplore.ieee.org/document/8743470/

[17] Y. Yang, W. Hong, and S. Li, "Deep ensemble learning based probabilistic load forecasting in smart grids," *Energy*, vol. 189, Dec. 2019, Art. no. 116324. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0360544219320195

[18] Y. Ye, D. Qiu, J. Li, and G. Strbac, "Multi-period and multi-spatial equilibrium analysis in imperfect electricity markets: A novel multi-agent deep reinforcement learning approach," *IEEE Access*, vol. 7, pp. 130515–130529, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8826539/

[19] Y. Wang, "Probabilistic individual load forecasting using pinball loss guided LSTM," *Appl. Energy*, vol. 235, pp. 10–20, Feb. 2019.

[20] X. G. Agoua, R. Girard, and G. Kariniotakis, "Probabilistic models for spatio-temporal photovoltaic power forecasting," *IEEE Trans. Sustain. Energy*, vol. 10, no. 2, pp. 780–789, Apr. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8385146/

[21] D. W. Van der Meer, J. Widén, and J. Munkhammar, "Review on probabilistic forecasting of photovoltaic power production and electricity consumption," *Renew. Sustain. Energy Rev.*, vol. 81, pp. 1484–1512, Jan. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1364032117308523

[22] Y. Gu, Q. Chen, K. Liu, L. Xie, and C. Kang, "GAN-based model for residential load generation considering typical consumption patterns," in *Proc. IEEE Power Energy Soc. Innov. Smart Grid Technol. Conf. (ISGT)*, Washington, DC, USA, Feb. 2019, pp. 1–5. [Online]. Available: https://ieeexplore.ieee.org/document/8791575/

[23] M. Loschenbrand, S. Gros, and V. Lakshmanan, "Generating scenarios from probabilistic short-term load forecasts via non-linear Bayesian regression," in *Proc. Int. Conf. Smart Energy Syst. Technol. (SEST)*, Vaasa, Finland, Sep. 2021, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/9543288/

[24] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *J. Mach. Learn. Res.*, vol. 14, pp. 1303–1347, 2013.

[25] M. Löschenbrand, "Stochastic variational inference for probabilistic optimal power flows," *Electr. Power Syst. Res.*, vol. 200, Nov. 2021, Art. no. 107465. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0378779621004466

[26] D. Cao, J. Zhao, W. Hu, Y. Zhang, Q. Liao, Z. Chen, and F. Blaabjerg, "Robust deep Gaussian process-based probabilistic electrical load forecasting against anomalous events," *IEEE Trans. Ind. Informat.*, [Online]. Available: https://ieeexplore.ieee.org/document/9435025/

[27] Z. Deng, B. Wang, Y. Xu, T. Xu, C. Liu, and Z. Zhu, "Multi-scale convolutional neural network with time-cognition for multi-step short-term load forecasting," *IEEE Access*, vol. 7, pp. 88058–88071, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8752362/

[28] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A generative model for raw audio," 2016, *arXiv:1609.03499*. [Online]. Available: http://arxiv.org/abs/1609.03499

[29] Y. Chen, Y. Kang, Y. Chen, and Z. Wang, "Probabilistic forecasting with temporal convolutional neural network," *Neurocomputing*, vol. 399, pp. 491–501, Jul. 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925231220303441

[30] L. Cheng, H. Zang, Y. Xu, Z. Wei, and G. Sun, "Probabilistic residential load forecasting based on micrometeorological data and customer consumption pattern," *IEEE Trans. Power Syst.*, vol. 36, no. 4, pp. 3762–3775, Jul. 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9324967/

[31] M. Imani, "Electrical load-temperature CNN for residential load forecasting," *Energy*, vol. 227, Jul. 2021, Art. no. 120480. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0360544221007295

[32] J. Qu, Z. Qian, and Y. Pei, "Day-ahead hourly photovoltaic power forecasting using attention-based CNN-LSTM neural network embedded with multiple relevant and target variables prediction pattern," *Energy*, vol. 232, Oct. 2021, Art. no. 120996. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0360544221012445

[33] H. Wang, H. Yi, J. Peng, G. Wang, Y. Liu, and H. Jiang, "Deterministic and probabilistic forecasting of photovoltaic power based on deep convolutional neural network," *Energy Convers. Manage.*, vol. 153, pp. 409–422, Dec. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S019689041730910X

[34] Q. Huang, J. Li, and M. Zhu, "An improved convolutional neural network with load range discretization for probabilistic load forecasting," *Energy*, vol. 203, Jul. 2020, Art. no. 117902. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0360544220310094

[35] H. Hao, Y. Wang, Y. Xia, J. Zhao, and F. Shen, "Temporal convolutional attention-based network for sequence modeling," 2020, *arXiv:2002.12530*. [Online]. Available: http://arxiv.org/abs/2002.12530

[36] L. Sehovac and K. Grolinger, "Deep learning for load forecasting: Sequence to sequence recurrent neural networks with attention," *IEEE Access*, vol. 8, pp. 36411–36426, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9006868/

[37] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, U. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[38] M. Afrasiabi, "Multi-agent microgrid energy management based on deep learning forecaster," *Energy*, vol. 186, Nov. 2019, Art. no. 115873.

[39] M. J. M. Al Essa, "Home energy management of thermostatically controlled loads and photovoltaic-battery systems," *Energy*, vol. 176, pp. 742–752, Jun. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0360544219306656

[40] A. S. N. Farsangi, S. Hadayeghparast, M. Mehdinejad, and H. Shayanfar, "A novel stochastic energy management of a microgrid with various types of distributed energy resources in presence of demand response programs," *Energy*, vol. 160, pp. 257–274, Oct. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0360544218312015

[41] Y. Wang, N. Zhang, Y. Tan, T. Hong, D. S. Kirschen, and C. Kang, "Combining probabilistic load forecasts," *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3664–3674, Jul. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8356127/

[42] N. M. M. Bendaoud, N. Farah, and S. Ben Ahmed, ''Comparing generative adversarial networks architectures for electricity demand forecasting,'' *Energy Buildings*, vol. 247, Sep. 2021, Art. no. 111152. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0378778821004369

[43] M. Sharifzadeh, A. Sikinioti-Lock, and N. Shah, ''Machine-learning methods for integrated renewable power generation: A comparative study of artificial neural networks, support vector regression, and Gaussian process regression,'' *Renew. Sustain. Energy Rev.*, vol. 108, pp. 513–538, Jul. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1364032119301807

[44] D. Bahdanau, K. Cho, and Y. Bengio, ''Neural machine translation by jointly learning to align and translate,'' 2014, *arXiv:1409.0473*. [Online]. Available: http://arxiv.org/abs/1409.0473

[45] D. Wingate and T. Weber, ''Automated variational inference in probabilistic programming,'' 2013, *arXiv:1301.1299*. [Online]. Available: http://arxiv.org/abs/1301.1299

[46] D. P Kingma and M. Welling, ''Auto-encoding variational Bayes,'' 2013, *arXiv:1312.6114*. [Online]. Available: http://arxiv.org/abs/1312.6114

[47] D. P. Kingma and J. Ba, ''Adam: A method for stochastic optimization,'' 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[48] K. Chen, K. Chen, Q. Wang, Z. He, J. Hu, and J. He, ''Short-term load forecasting with deep residual networks,'' *IEEE Trans. Smart Grid*, vol. 10, no. 4, pp. 3943–3952, Jul. 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8372953/

[49] L. Mescheder, S. Nowozin, and A. Geiger, ''Adversarial variational Bayes: Unifying variational autoencoders and generative adversarial networks,'' 2017, *arXiv:1701.04722*. [Online]. Available: http://arxiv.org/abs/1701.04722

[50] S. E. Razavi, A. Arefi, G. Ledwich, G. Nourbakhsh, D. B. Smith, and M. Minakshi, ''From load to net energy forecasting: Short-term residential forecasting for the blend of load and PV behind the meter,'' *IEEE Access*, vol. 8, pp. 224343–224353, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9292948/

[51] S. Wang, X. Wang, S. Wang, and D. Wang, ''Bi-directional long short-term memory method based on attention mechanism and rolling update for short-term load forecasting,'' *Int. J. Elect. Power Energy Syst.*, vol. 109, pp. 470–479, Jul. 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0142061518328047

MARKUS LÖSCHENBRAND received the M.Sc. degree in supply chain management from the Vienna University of Economics and Business (WU Vienna), in 2015, and the Ph.D. degree from the Norwegian University of Science and Technology (NTNU), Trondheim, Norway, in 2019. He is currently a Research Scientist with SINTEF Energy Research, Trondheim. His research interests include renewable energy, economics, game theory, optimization, probabilistic programming, and machine learning.