Short-Term Density Forecasting of Low-Voltage Load Using Bernstein-Polynomial Normalizing Flows

Marcel Arpogaus, Marcus Voss, Beate Sick, Mark Nigge-Uricher, and Oliver Dürre

Abstract—The transition to a fully renewable energy grid requires better forecasting of demand at the low-voltage level to increase efficiency and ensure reliable control. However, high fluctuations and increasing electrification cause huge forecast variability, not reflected in traditional point estimates. Probabilistic load forecasts take uncertainties into account and thus allow more informed decision-making for the planning and operation of low-carbon energy systems. We propose an approach for flexible conditional density forecasting of short-term load based on Bernstein polynomial normalizing flows, where a neural network controls the parameters of the flow. In an empirical study with 3639 smart meter customers, our density predictions for 24h-ahead load forecasting compare favorably against Gaussian and Gaussian mixture densities. Furthermore, they outperform a non-parametric approach based on the pinball loss, especially in low-data scenarios.

Index Terms—Normalizing flows, probabilistic regression, deep learning, probabilistic load forecasting, low-voltage.

I. INTRODUCTION

A. Motivation

On THE path to a sustainable energy supply, the take-up of renewable and distributed energy resources transforms the electric energy system into a more decentralized system. This increases the role of low-voltage (LV) grids that typically make up the largest part of distribution systems but are still the least monitored and controlled. Regulations such as the German “Klimaschutzgesetz” or “U.K. Green Industrial Revolution: 10 Point Plan” and their consequent funding programs promote the electrification of mobility, heating, and industrial process, leading to rapid changes in electricity generation and consumption – to the extent that they already cause significant changes in load curves and power flows in the distribution network [1].

Hence, accurate local short-term load and generation forecasts at the LV level ranging from minutes to days ahead are becoming essential for grid operators, utilities, building- and district operators, and the customers themselves to make informed decisions within many applications. Such applications include, for instance, peak load reduction [2] and voltage control [3]. Accurate load forecasts can also be used for grid state estimation [4]. They can also be used for anomaly detection to increase resiliency for the grid [5] or to detect energy theft [6]. Short-term forecasts can further inform participants of local and peer-to-peer energy markets [7], real-time pricing schemes [8] or flexibility applications [9] that are emerging with the energy transition. See [10] and [11] for reviews on more applications.

There are a plethora of load forecasting approaches that have been proposed on the aggregated and system-level. While most approaches are focused on point forecasts, a trend towards probabilistic approaches can be seen (see, for instance, as reviewed in [12], [13], [14]). While usual point forecasts model the expected value, probabilistic forecasts model the distribution, enabling more informed decision-making by considering uncertainties. This makes probabilistic forecasts especially relevant for the LV level, as they can better handle the higher volatility present at this level when compared to the system-level [10], [13].

B. Related Work

Probabilistic forecasting at the LV level is challenging, as the time series are volatile, multivariate, and the marginal distributions are typically skewed and multi-modal [15]. Generally, uncertainty in load forecasts can be modelled as either simple prediction intervals [16], quantile estimates [17], [18], [19], [20] (providing estimates for a fixed set of quantiles), full continuous distributions [21], [22] or scenarios [23], see [10] for a review.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSG.2023.3254890.

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Further, the methods applied can be distinguished between statistical and time series models that often rely on simple parametric assumptions and models from the machine learning or deep learning domain, which can model complex dependencies at the cost of interpretability. Typical statistical models, capable of forecasting intervals and quantiles, are, for example, kernel regression [24], Gaussian modelling [16] and additive quantile regression [25]. Statistical models for the continuous distribution are the ARMA-GARCH [26] models. In deep learning, it is straightforward to model the full density distribution, with complex dependencies on explanatory variables in the input data. In this case, the distribution parameters are modelled by the last layer of a neural network. The complete network can then simply be trained with state of the art stochastic gradient descent and the standard loss function defined by the maximum likelihood principle [27]. So far, only rather simple parametric DL approaches have been proposed in the LV context, which make strong assumptions about the underlying distribution, such as a single Gaussian distribution [28], [29] or Gaussian Mixture Models (GMM) [30]. The modelling for more complex distributions in the context of LV level load forecasts has been done so far mostly indirectly via quantile regression for different NN architectures: Neural Networks (NN) [3], [31], [32], Residual Networks [20], Long Short-term Memory (LSTM) [17], [33] and Convolutional Neural Networks (CNN) [19]. These methods cannot be minimized using the standard maximum likelihood principle and, instead rely on an ad-hoc loss like the pinball loss. Further, care has to be taken to avoid quantile crossing.

C. Contribution of This Work

Here, we propose a flexible method which fits directly into the deep learning framework. It is based on Normalizing Flows (NFs) flexible parameterized transformations from a simple distribution (e.g., Gaussian) to an arbitrary complex one (see [34] for an overview). Up to now, NF models have gained the most attention in applications where complex high-dimensional unconditioned distributions \( p_y(y) \) are modelled, i.e., image generation [35] or speech synthesis [36]. Probabilistic regression models based on NF, modelling the CPD \( p_y(y|x) \) have gained little attention. However, recent research extended NF to conditional density estimation with very promising results [37], [38], [39] and applied it to time-series forecasting [40] and scenario generation of residential loads [41], [42].

We propose an approach based on Bernstein-Polynomial Normalizing Flows (BNFs) for short-term density forecasting of LV loads, which have been used in the statistics community for a while [43] and were brought to deep learning by [39]. Compared to existing methods, BNF allow full density forecasts without strong parametric assumptions on the distribution and flexible modelling of explanatory variables. This makes the approach especially suitable for the challenging LV level. Our empirical results demonstrate that the proposed method is capable of fitting a single model for multiple customers, rather than requiring a separate model for each individual customer. Moreover, our results indicate that models using deep artificial neural networks to estimate density parameters can generalize well to previously unseen but similar situated households.

This work extends an earlier preliminary version [44], with an in-depth discussion of the developed method and a more rigorous comparison with the other methods. The robustness has been confirmed by the stable test performances of 10 fits of the model, each trained on the same train data after random initialization.

In Section II we give a brief introduction to NFs and describe our BNF approach for short-term density forecasting of LV loads and highlight the differences to previous BNF implementations. In Section III we use public data from 3,639 smart meter customers of the CER dataset [45] to compare two different NN architectures combined with four parametric and non-parametric density estimation methods in 24h-ahead forecasting. Finally, we conclude this study in Section IV.

II. Normalizing Flows Using Bernstein-Polynomials

We tackle the load forecasting problem in the framework of deep probabilistic regression. For the covariates \( x \) such as lagged power consumption at earlier time steps, holiday indicator, or calendar variables, the Conditional Probability Distributions (CPDs) \( p_y(y|x) \), of the electric load at time-step \( t \) are predicted (Fig. 3). For notational convenience, we drop the subscript \( y \) and \( t \) when there is no ambiguity.

A. Background on Normalizing Flows (NF)

The main idea of NF is to fit a parametric bijective conditional function \( z = f(y|\theta(x)) \) that transforms between a possibly complex conditional target distribution \( p_y(y|x) \) and a simple distribution \( p_z(z) \), often \( p_z(z) = N(0, 1) \). The change of variable formula allows us to calculate the probability \( p_y(y|x) \) from the simple probability \( p_z(z) \) as follows:

\[
\begin{align*}
p_y(y|x) = p_z(z) |\det \nabla f(y|\theta(x))| \end{align*}
\]

With the Jacobian determinant \( \det \nabla f(y|\theta(x)) \) ensuring that the probability integrates to one after the transformation (hence the name normalizing flow). Sampling from the learned data distribution \( p_z \) is then achieved by first sampling \( z \) from the simple base distribution \( p_z \) and passing it through the inverse transformation \( f^{-1}(z|\theta(x)) \) to obtain \( y \). This leads to the two main properties the transformation functions must satisfy: i) \( f \) must be invertible, usually guaranteed by restricting the transformation function to be strictly monotone, and ii) both \( f \) and \( f^{-1} \) must be differentiable [34].

The parameters \( \theta \) of the bijective transformation functions \( f \) can then be tuned by maximizing the likelihood \( L \) of the samples \( y \), observed under condition \( x \) via Eq. (1). For numerical reasons, it is common practice to minimize the negative Negative Logarithmic Likelihoods (NLLs) function instead

\[
NLL = -\sum_{y|x \in D} \log(p_y(y|x, \theta)) \]

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Probabilistic Regression With Bernstein Flows

Most of the recent NF methods in the Machine Learning literature construct the flow from a combination of $K$ simple transformation functions $f_i$, to compose more expressive transformations $f(z) = f_K \circ f_{K-1} \circ \cdots \circ f_1(y)$ [34]. Chaining many of those simple transformations can result in a flexible bijective transformation. Alternatively, there are approaches that use a single flexible transformation, such as sum-of-squares polynomials [46], or splines [47], [48]. Our approach benefits from Bernstein polynomials introduced recently in the statistics community [43] and combined with NN in [39], [49]. Compared to other methods Bernstein-Polynomial Normalizing Flow (BNF) have several advantages, like 1) robustness against initial and round-off errors, 2) higher interpretability, 3) a theoretical upper bound for the approximation error, 4) ability to increase the flexibility at no cost to the training stability. See [49] for an in-depth theoretical analysis of the framework.

Our implementation extends the work of Sick et al. [39] but has some significant improvements to enhance convergence and allow a stable inversion of the flow from the latent variable $z$ to the observed $y$, which is needed, i.e., for sampling new load profiles from the learned distributions (Fig. 8).

The complete NF $f = f_2 \circ f_1$ consists of two functions, $f_1$ and $f_2$, as shown in Fig. 1. The first transformation $f_1 : z_1 = a_1(x) \cdot y - b_1(x)$, allows transforming $y$ into the domain $[0, 1]$ of the Bernstein polynomial $f_2$ of order $M$ defined as:

$$f_2(z_1) = \frac{1}{M+1} \sum_{i=0}^{M} \text{Be}_i(z_1) \vartheta_i \tag{3}$$

Generated by the $M + 1$ densities of the Beta distribution $\text{Be}_i(z) = f_{i+1,M-i+1}(z)$, multiplied with the corresponding Bernstein coefficients $\vartheta_0, \ldots, \vartheta_M$. Bernstein polynomials were first introduced by Bernstein [50] to prove the Weierstrass approximation theorem. This theorem states that every continuous function on any fixed interval can be approximated arbitrarily well by a polynomial with sufficient order. Bernstein introduced the Bernstein polynomials for a constructive proof by showing that they can approximate every function in $f : z \in [0, 1] \rightarrow \mathbb{R}$ for $M \rightarrow \infty$ (see [51] for details of the proof and further properties of the Bernstein polynomials). Higher degree Bernstein polynomials increase the expressiveness with no cost to the training stability [43], [49]. Empirically, $M \gtrsim 10$ polynomials are often sufficient in typical regression settings [43].

The Bernstein polynomials are bounded between $z \in [0, 1]$. In [39] this has been ensured by a sigmoid function, but when evaluating the inverse flow, we found some serious drawbacks in numerical stability. Hence, we decided to remove the sigmoid function and instead do a linear extrapolation of $f_2$ for $z \notin [0, 1]$. Moreover, [39] used a second affine transformation $f_3$. Since we could not find a beneficial effect (w.r.t. NLL) of $f_3$, we removed it completely.

To ensure the invertibility of $f$, we choose the individual transformations to be bijective by requiring strict monotonicity. For $f_1$ we need to ensure that the scale parameter $a_1$ is positive. This is done by applying a softplus activation function to the output of the network (Fig. 2). The required monotonicity of $f_2$ can be easily ensured by enforcing an increasing ordering of the Bernstein coefficients $\vartheta_0, \ldots, \vartheta_M$.

The easiest way to archive this is by recursively applying a strictly positive real function such as softplus to an unconstrained and unordered vector $\tilde{\vartheta}_0, \ldots, \tilde{\vartheta}_M$, such that $\vartheta_0 = \tilde{\vartheta}_0$ and $\vartheta_k = \tilde{\vartheta}_{k-1} + \text{softplus}(\tilde{\vartheta}_k)$ for $k = 1, \ldots, M$. In this simple approach, the convergence in parameter estimation depends on parameter initialization, since $\vartheta_0$ is derived directly from the unconstrained parameters.

We thus require that the transformation $f_2$ covers at least the range $[-3, 3]$, i.e., $\pm 3\sigma$ of the standard Gaussian. Since the boundaries of Bernstein polynomials are given by their first and last coefficient $f_2(0) = \vartheta_0$ and $f_2(1) = \vartheta_M$, we can determine these values from unrestricted parameters $\tilde{\vartheta}_0$ and $\tilde{\vartheta}_M$ via $\vartheta_0 = -\text{softplus}(\tilde{\vartheta}_0) - 3.0 \leq -3$ and $\vartheta_M = \text{softplus}(\tilde{\vartheta}_M + 3.0 \geq 3$. To ensure $\sum_{k=1}^{M} (\vartheta_k - \vartheta_{k-1}) = \vartheta_M - \vartheta_0 =: \Delta$ the remaining coefficients $\vartheta_k$ for $k = 1, \ldots, M$ can be determined as:

$$\vartheta_k = \vartheta_{k-1} + \Delta \cdot \text{softmax}\left(\begin{bmatrix} \tilde{\vartheta}_1, \tilde{\vartheta}_3, \ldots, \tilde{\vartheta}_M \end{bmatrix} \right)_{k-1} \tag{4}$$
TABLE I
THE DATASET [45] WAS SPLIT BY CUSTOMERS AND DATE-RANGES INTO ONE TRAIN AND THREE TEST SETS. ONLY THE TRAIN DATA WAS USED TO OPTIMIZE THE WEIGHTS OF THE NNS

| date range       | households used in training | hold-out |
|------------------|----------------------------|----------|
| 14/07/2009 – 31/07/2010 | Train                     | Test 2   |
| 01/08/2010 – 31/12/2010 | Test 1                    | Test 3   |

since \( \Delta \) and all components of the softmax are non-negative, \( \theta_k \geq \theta_{k-1} \geq 0 \) so that \( f_2 \) is monotonous as required. To sample new data from a learned distribution, the inversion of \( f_2 \) is required. Since there is no closed-form solution for the inversion of higher-order Bernstein polynomials, a root-algorithm is used to determine the inverse [52].

Altogether, the complete function \( f \) has the following \( M + 3 \) parameters \( \theta = (a_1(\mathbf{x}), b_1(\mathbf{x}), \theta_0(\mathbf{x}), \ldots, \theta_M(\mathbf{x})) \), obtained from the \( M + 4 \) unconstrained outputs of a NN, as illustrated in Fig. 2.

Training is done by optimizing the weights of the NN, which controls the parameters of the chained transformations \( f_2 \circ f_1 \) w.r.t. the NLL (Eq. (2)) of the training data points, using Eq. (1). All models have been trained with the Adam optimizer [53] and early stopping with a maximum of 300 epochs, to prevent over-fitting. Additionally, the learning rate was reduced by a factor of ten after every three epochs without significant improvement of the validation loss.

III. LOAD FORECASTING SIMULATION STUDY

This section presents the evaluation of the BNF approach for load forecasting in an empirical study. First, the used dataset is described, then the forecasting approach and benchmark methods are introduced, and finally, the results are discussed.

A. Dataset Description

The models were trained on a dataset containing information on the electricity demand of smart meter customers in Ireland recorded at a resolution of 30 minutes during the period from 14/07/2009 to 31/12/2010 [45]. During preprocessing, all non-residential (2, 220) buildings were dropped, since the stochastic behavior of residential customers (4, 225) was of explicit interest in this study. In addition, all incomplete records (586) were removed so that the final data set includes \( N = 3,639 \) of the original 6, 445 customers.

To obtain the training data, measurements up to 07/31/2010 23:30 were selected from a random subset of size \( N_{\text{train}} \) of all \( N \) households.\(^1\) This training data is used to determine the weights of the neural network-based approaches. For testing, there exist two approaches of practical relevance. We can either ask how a model trained on \( N_{\text{train}} \) households performs in the future on the same households (Test 1) or on new, yet unseen, households (Test 3). For completeness, we also include the less relevant case (Test 2), i.e., trained and evaluated in the same period but for different households. An Overview of these data splits is given in Table I.

The load data has then been normalized into the range \([0, 1]\). Since the exact location of different assets is unknown, no additional data like whether variables have been used. The scripts used to preprocess the data set and conduct the experiments are available on GitHub.\(^2\)

At runtime, the data is shuffled and batched into mini batches of size 1,024. Each sample consists of an input tuple \( \mathbf{x} = (x_h, x_m) \) containing the historical data \( x_h \), with the lagged electric load of the past seven days and meta data \( x_m \), with trigonometric encoded time information (see [54]) and a binary holiday indicator as indicated in Fig. 3. The prediction target \( \mathbf{y} \) is the load for the next day, with resolution of 30 minutes. Hence, the models predict 48 CPDs \( p(y_1|\mathbf{x}), \ldots, p(y_{48}|\mathbf{x}) \) for every future time step.

B. Probabilistic Forecasting Models

In this study, we used 4 different model types (BNF, GMM, GM, QR), which are explained in the following, to forecast the Conditional Probability Distribution (CPD) of the load for 48 time steps. To control the parameters of the forecasting models, we used two NN architectures, a \textit{fully connected neural network} (FC) and a \textit{dilated 1D-Convolution neural network} (1DCNN) (Fig. 3). Architecture choices were made based on prior experience and known best practices [55], [56] and did not undergo further optimizations.

The FC is configured with three hidden layers, with 512, 256 and finally 128 units using the ELU activation function [57]. The historical data was flattened and concatenated with the metadata. This model is not specifically designed for processing the sequential data, but serves as a baseline model to assess the influence of a more sophisticated NN model.

The 1DCNN is inspired by the WaveNet architecture [58]. The model was built by stacking 8 1D convolutional layers with doubling dilatation rates \( 1, 2, 4, \ldots, 128 \), this principle is illustrated in Fig. 4. This results in a model with a receptive field capturing 256 input values. Hence, almost the whole input sequence, consisting of one week of historical load data with a total of 348 input features, can be processed at once. Each of these dilated convolutions uses the ReLU activation function and has 20 filters. Finally, a regular 1D convolutional layer without dilatation, again ReLU activation and 10 filters. The output of this last convolutional layer is then flattened and concatenated with the metadata before it is fed into a final fully connected layer with ELU activation function, followed by a last layer with a linear activation function to generate the output.

\(^1\)The last 10\% of that period were used for validation during development.

\(^2\)https://github.com/MArpogaus/stplf-bnf
For predicting the CPDs this study proposes the Bernstein-Polynomial Normalizing Flow (BNF) in which $\theta$ are the parameters of the Bernstein polynomial as well as the additional linear transformations making up the flow $f$ (Fig. 1). We choose 20 outputs for the NNs, hence the Bernstein polynomials are of order $M = 16$. Higher orders polynomials generally lead to more flexible models, but the improvements are diminishing for high $M$ [43]. We compare the BNF with three benchmarks to model the CPD, a simple Gaussian Model (GM), a Gaussian Mixture Models (GMM), and as a non-parametric approach, a Quantile Regression (QR).

The simple Gaussian Model (GM) is a probabilistic extension of regular regression, predicting not only the conditioned mean $\mu(x)$, but also the conditional variance $\sigma^2(x)$ of a Gaussian distribution [28], [29], [59].

To model more complex distribution shapes, e.g., with multiple modes, a Gaussian Mixture Models (GMM) of three normal distributions was implemented. The output vector $\theta$ contains the mean and variance $\mu_k(x), \sigma_k^2(x)$ and the mixing coefficients $\alpha_k(x)$ for $k = 1, 2, 3$ (see, e.g., [30]).

The Quantile Regression (QR) is a typical baseline in probabilistic load forecasting (cf., e.g., [17], [19]). It is configured to predict the 99 quantiles $p = 0.01, \ldots, 0.99$ for each time step, which have been constrained to be monotonically increasing by applying a softplus activation function and then calculating the cumulative sum, to prevent quantile-crossing. Note that the QR is not a continuous CPD, hence the NLL is not tractable, and instead the pinball loss (Eq. (6)) is minimized.

As a naive baseline, we use the measurements of all households for the respective time-point in the training period to determine the Empirical Cumulative Density Function (ECDF).

Table II summarizes the number of trainable parameters for each model and its output shape.

### C. Proper Scoring Rules

Recently, reviews have repeatedly noted that there is a lack of common evaluation methods in the literature on load forecasting, which makes it difficult to compare different methods [10], [13], [14]. In classical forecasting literature, it is well established to assess probabilistic predictions in terms of “sharpness of the predicted density, subject to calibration” [60]. In this context, calibration or reliability refers to the statistical consistency between the predicted densities and observations. Sharpness describes the property of predicted density to concentrate tightly around the actual outcome. In practice, proper scoring rules allow for assessing both sharpness and calibration simultaneously [61]. A scoring rule is proper, if its value is minimized if the predicted density equals the real distribution and strictly proper if this minimum is global, hence only reached if the prediction equals the real distribution [60]. We follow these recommendations and evaluate the strictly proper NLL (Eq. (2)) along with two well established proper scoring rules introduced in the following.

The Continuous Ranked Probability Score (CRPS) is, besides the NLL, one of the most common proper scores to evaluate the performance of probabilistic forecasts. It evaluates the quality of the predicted cumulative distribution function and is defined as [62], [63]:

$$\text{CRPS}(F, y) = \frac{1}{N} \sum_{i=1}^{N} \int_{-\infty}^{\infty} (F_i(x) - 1(y_i \leq x))^2 dx$$

Here $F_i$ corresponds to the predicted cumulative density function for time-step $t$ and $1$ is the unit step function.

To quantify the accuracy of quantile forecasts, often the Mean Quantile Score (MQS) is used with

$$\text{MQS}(y, q_p, p) = \frac{1}{N} \sum_{i=1}^{N} (q_i - y_i) \mathbf{1}(y_i \leq q_i) - p$$

where $q_i$ is the forecast for the $p$-quantile at time-step $t$ where $y_i$ was observed [63]. It is equivalent to the Pinball Loss of the QR model [17]. When predicting continuous distributions or multiple quantiles $q$, there exists a formulation which can be seen as a discrete variant of CRPS [64]

$$\text{CRPS}(y, q_p, p) \approx 2\Delta p \sum_{y \in q} \text{MQS}(y, q_p)$$

Here the quantile distance $\Delta p$ is assumed constant. We used the latter, to allow comparability to our CRPS implementation. Because both MQS and CRPS are scale-dependent, we calculated these metrics using the normalized data for ease of comparison across different datasets and will refer to them as normalized MQS (NMQS) and normalized CRPS (NCRPS) and report them as percentages [12], [65].
D. Qualitative Results

Fig. 5 gives an example of a load forecast for the Irish Christmas holidays for an individual household in Ireland. The load profile has distinct peaks that correspond to specific activities within the household and exhibits an extraordinarily high consumption compared to the rest of the year. The plot displays the uncertainty information in the form of confidence intervals for 98% and 60%. This example forecasts in Fig. 5 demonstrate how probabilistic forecasts can handle the specific characteristics of high volatility and complex influences by explicitly modelling the uncertainty, information that is intractable with point forecasting methods.

The two neural networks FC and 1DCNN lead to very different results. In the given examples, the FC variants tend to generate sharper predictions, the CI is smaller, and the predicted PDF is concentrating tighter around their mean. However, the FC models are seemingly less reliable and generate more noisy prediction compared to 1DCNN, especially for days with unusual high load. Most models’ tails reach far into the negative domain. We observed that our BNF model generates the most “realistic” distributions without additional constraints or assumptions about its shape.

E. Dependence on the Training Data Size

To assess the performance of the individual models w.r.t. the number of households $N_{\text{train}}$ in the training data, we trained all models on different subsets sizes $N_{\text{train}}$. The results are shown in Fig. 6.

The evaluation results reveal two things. First, the models’ performance appears to asymptotically approach a limit. Beyond a certain point (here about $N_{\text{train}} = 1091$ (30%)) they do not seem to improve further even when adding more households. Surprisingly, the differences for the individual test sets seems negligible even for small $N_{\text{train}}$. It is worth noting that the performance of all models on unknown households in the same period as the training data (Test 2) is always better than on future data (Test 1 and 3). Predictions of future events for known houses (Test 1) always outperform the prediction for unknown houses (Test 3). This allows us to conclude that in our case seasonal fluctuations are clearly more relevant than individual behavior patterns, and more households only slightly improve performance on unknown data. However, it remains unclear whether this is universally true and how future electricity demand will be affected by random events such as electric vehicle charging or rooftop photovoltaics.

Second, the performance of each model in the very data-poor settings $N_{\text{train}} \leq 363$ (10%) differs extremely. For the BNF, ECDF, and GMM, we see an almost identical, nearly linear decrease in NMQS, but with unexpectedly good results even for $N_{\text{train}}$ as low as 36 (1%). The GM and QR models seem to lose their predictive quality completely and end up with a NMQS outside the plot range. The 1DCNN version of QR achieves at best a NMQS of 2, 259.

F. Quantitative Comparison of Models

To assess the robustness, all the models have been fitted with 10 different random weight installations for $N_{\text{train}} \in \{363 (10\%), 1,091 (30\%)\}$ customers in the dataset. The median and standard error for the evaluation on data split Test 3 (Table I) are shown in Table III. Fig. 7 shows the spread for MQS of these individual runs in a box plot.

When looking at the results from Table III and Fig. 7, it becomes apparent that the GM model has the worst performance in NLL and NCRPS and more data only slightly improves
Fig. 6. Evaluation results of models trained on a different number of households for $N_{\text{train}} \in \{36 (1\%), 181 (5\%), 363 (10\%), 727 (20\%), 1,091 (30\%), 1,455 (40\%), 1,819 (50\%)\}$. Shown are the results of NMQS on the three test sets (see Table I) along with their mean. The dotted line represents the result of the ECDF baseline model. Lower is better for all values.

it. For the NLL it is even worse than the ECDF baseline. This illustrates how important it is to pick a density model, which is flexible enough to represent the complex nature of the data. Interestingly, the scores are not always better the more flexible the CPD is. We see that the median of NMQS is always lower for the flexible BNF and QR model than for both Gaussian based models GM and GMM, when trained on a $N_{\text{train}} = 1,091$ households. However, in the low-data scenario, with $N_{\text{train}} = 363$ QR falls behind the GMM for both NN architectures and the FC variant only converges for two of the ten runs. This confirms our finding that QR requires a solid amount of data to work, and the robustness of BNF.

Furthermore, the NN architecture has an substantial influence on the performance of the GMM and QR models, especially the 1DCNN version of the latter performs far better in the low-data setting. However, the BNF model achieves the overall best scores when trained on $N_{\text{train}} = 363$ households and performs as well as QR for $N_{\text{train}} = 1,091$. We speculate that the performance gain of the BNF compared to the QR in the low-data setting is because the BNF is less prone to overfitting since using the Bernstein basis is smoother compared to estimating 99 quantiles, and requires fewer parameters Table II. An additional benefit of the BNF over the QR is that it provides a continuous distribution. This confirms the superiority of the BNF approach, being flexible, stable and requiring less fine-tuning to achieve good results.

| $N_{\text{train}}$ | NN | kind Distribution | NLL | NCRPS (%) | NMQS (%) |
|-------------------|----|-------------------|-----|-----------|----------|
| 363               | FC | BNF              | -111.702 | 1.920     | 1.900    |
|                   | GMM | BNF              | -135.616 (+/-0.358) | 1.502 (+/-0.007) | 1.468 (+/-0.007) |
|                   | GM  | BNF              | -129.683 (+/-0.642) | 1.543 (+/-0.008) | 1.526 (+/-0.008) |
|                   | QR  | BNF              | -100.973 (+/-0.893) | 1.745 (+/-0.015) | 1.724 (+/-0.014) |
| 1091              | FC | BNF              | -135.262 (+/-0.361) | 1.443 (+/-0.009) | 1.428 (+/-0.009) |
|                   | GMM | BNF              | -135.029 (+/-0.734) | 1.464 (+/-0.009) | 1.449 (+/-0.009) |
|                   | GM  | BNF              | -104.128 (+/-0.402) | 1.660 (+/-0.010) | 1.642 (+/-0.010) |
|                   | QR  | BNF              | -104.776 (+/-0.906) | 1.341 (+/-0.013) | 1.323 (+/-0.014) |
|                   | -   | BNF              | -103.355 (+/-1.027) | 1.559 (+/-0.022) | 1.541 (+/-0.021) |

IV. CONCLUSION

Forecasting at the LV level is becoming essential for many stakeholders while more and more applications in low carbon energy systems are explored. Due to the high volatility of load profiles, probabilistic load forecasts are an emerging research topic, as they are capable of expressing uncertainties introduced by the fluctuations caused by the increasing penetration of renewable and distributed energy sources. The majority of probabilistic load forecasting literature focuses on parametric or QR approaches for estimating marginal distributions. Parametric methods need to make assumptions about the underlying distribution and so cannot model complex distributions. QR based models, on the other hand, can only provide a discrete approximation of the full distribution and thus inherently have the following limitation: They cannot provide density estimates for arbitrary quantiles, sampling from them is not straightforward, and they do not fit into the maximum likelihood-based deep learning framework. Instead, the proposed BNF model uses a cascade of parameterized transformation functions, known as Normalizing Flows (NFS), to model the probability density. Model parameters can be obtained by minimizing the NLL through gradient descent and hence fit perfectly into the standard deep learning approach.

We demonstrated that BNFs are a powerful and stable method to express complex non-Gaussian distributions, with almost no regularization or special tuning. When there is little data available BNFs performed better than QR, while for larger amounts of training data, both methods are practically en par. This makes BNFs a preferential choice over the QRs or Gaussian approaches for probabilistic load forecasts.
A possible enhancement might be to take the multivariate nature of the forecast more directly into account. Instead of predicting multiple marginal CPDs for a fixed forecast horizon, a future implementation could benefit from autoregressive architectures for non-fixed forecast horizons or extending the BNF to multivariate versions and use it, i.e., for sampling new realistic load profiles (see exemplarily in Fig. 8). This makes BNFs also applicable for other use cases like generation of synthetic scenarios for grid planning or stochastic optimization approaches. In addition, ensembles have demonstrated superior predictive performance in probabilistic load forecasting [31], thus recent transformation ensembles [66] are a promising extension of the current framework to improve both performance and interpretability.

ACKNOWLEDGMENT

Public data from the CER Smart Metering Project accessed via the Irish Social Science Data Archive (ISSSDA) were used in the development of this project [45].

The authors would like to thank the reviewers for their valuable and detailed comments.

REFERENCES

[1] F. Heymann, J. Silva, V. Miranda, J. Melo, F. J. Soares, and A. Padilha-Feltrin, "Distribution network planning considering technology diffusion dynamics and spatial net-load behavior," Int. J. Electr. Power Energy Syst., vol. 106, pp. 254–265, Mar. 2019, doi: 10.1016/j.ijepes.2018.10.006.

[2] M. Rowe, T. Yunusov, S. Haben, C. Singleton, W. Holderbaum, and B. Potter, "A peak reduction scheduling algorithm for storage devices on the low voltage network," IEEE Trans. Smart Grid, vol. 5, no. 4, pp. 2115–2124, Jul. 2014, doi: 10.1109/TSG.2018.2799903.

[3] T. Zufferey, S. Renggli, and G. Hug, "Probabilistic state forecasting and optimal voltage control in distribution grids under uncertainty," Electr. Power Syst. Res., vol. 188, Nov. 2020, Art. no. 106562, doi: 10/gn6xqx.

[4] J. Hermanns et al., "Evaluation of different development possibilities of distribution grid state forecasts," Energies, vol. 13, no. 8, p. 1891, 2020, doi: 10/gkwc9.

[5] Z. M. Fadlullah, M. M. Fouda, N. Kato, X. Shen, and Y. Notzaki, "An early warning system against malicious activities for smart grid communications," IEEE Netw., vol. 25, no. 5, pp. 50–55, Oct. 2011, doi: 10/0fpfx28.

[6] G. Fenza, M. Gallo, and V. Loia, "Drift-aware methodology for anomaly detection in smart grid," IEEE Access, pp. 7, pp. 9645–9657, 2019, doi: 10/gknwc6.

[7] T. Morstyn, N. Farrell, S. J. Durby, and M. D. McCulloch, "Using peer-to-peer energy-trading platforms to incentivize prosumers to form federated power plants," Nat. Energy, vol. 3, no. 2, pp. 94–101, 2018, doi: 10.1038/s41560-017-0075-y.

[8] B. He, J. Li, F. Tsung, Y. Gao, J. Dong, and Y. Dang, "Monitoring of power consumption requirement load process and price adjustment for smart grid," Comput. Ind. Eng., vol. 137, Nov. 2019, Art. no. 106068, doi: 10.1016/j.cie.2019.106068.

[9] J. Ponocko and J. V. Milanović, "Forecasting demand flexibility of aggregated residential load using smart meter data," IEEE Trans. Power Syst., vol. 33, no. 5, pp. 5446–5455, Sep. 2018, doi: 10.1109/TPWRS.2018.2799903.

[10] S. Haben, S. Arora, G. Giasemidis, M. Voss, and D. Vukadinović Greetham, "Review of low voltage load forecasting: Methods, applications, and recommendations," Appl. Energy, vol. 304, Dec. 2021, Art. no. 117798, doi: 10.1016/j.apenergy.2021.117798.

[11] 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, doi: 10.1109/TSG.2018.2818167.

[12] 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, doi: 10.1016/j.rser.2017.05.212.

[13] T. Hong, P. Pinson, Y. Wang, R. Weron, D. Yang, and H. Zareipour, "Energy forecasting: A review and outlook," IEEE Open Access J. Power Energy, vol. 7, pp. 376–388, 2020, doi: 10/gn6xq4.

[14] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," Int. J. Forecast., vol. 32, no. 3, pp. 914–938, 2016, doi: 10.1016/j.ijforecast.2015.11.011.

[15] M. Anvari, E. Procidrou, B. Schaefer, C. Beck, H. Kantz, and M. Timme, "Data-driven load profiles and the dynamics of residential electric power consumption," 2020, arXiv: 2009.09287.
[63] J. W. Messner, P. Pinson, J. Browell, M. B. Bjerregård, and I. Schicker, “Evaluation of wind power forecasts—An up-to-date view,” Wind Energy, vol. 23, no. 6, pp. 1461–1481, 2020, doi: 10/gg47c3.

[64] F. Laio and S. Tamea, “Verification tools for probabilistic forecasts of continuous hydrological variables,” Hydrol. Earth Syst. Sci., vol. 11, no. 4, pp. 1267–1277, 2007, doi: 10/fndsbk.

[65] S. Haben, G. Giasemidis, F. Ziel, and S. Arora, “Short term load forecasts of low voltage demand and the effects of weather,” Int. J. Forecast., vol. 35, no. 4, pp. 1469–1484, 2019, doi: 10/gn8rk2.

[66] L. Kook, A. Götschi, P. F. Baumann, T. Hothorn, and B. Sick, “Deep interpretable ensembles,” 2022, arXiv: 2205.12729.

**Marcel Arpogaus** received the B.Eng. degree in electrical engineering and the M.Sc. degree in computer science from the Konstanz University of Applied Science (UAS) in 2020. He is currently working as a Research Associate in the Research Project “AI4Grids” with UAS, where he is developing machine and deep learning algorithms for energy forecasting, distribution grid operation, and planning.

**Marcus Voss** received the M.Sc. degree in business information systems from the Humboldt University of Berlin in 2014. He is currently pursuing the Ph.D. degree with TU Berlin. He was a Research Associate with the Distributed Artificial Intelligence Laboratory, TU Berlin, leading the research group “AI4Smart.” He is an AI Expert with Birds on Mars GmbH. In his Ph.D. research, he focuses on analyzing low-voltage smart meter data using machine learning.

**Beate Sick** received the Ph.D. degree in experimental physics from ETHZ. Then, she turned to biostatistics and was responsible for the bioinformatics at the DNA Array facility of UNIL and EPFL before joining ZHAW. She is a Professor of Applied Statistics with the Zurich University of Applied Science and co-affiliated with the University of Zurich. Her current main research focus is on leveraging deep learning approaches for medical research.

**Mark Nigge-Uricher** received the M.Sc. degree in global innovation management from the University of Strathclyde and Aalborg University in 2012. He is a Senior Managing Consultant with Bosch.IO GmbH, where he was responsible for the low-voltage energy management domain, mainly related to virtual power plants. Additionally, he consults in IoT and business model innovation projects.

**Oliver Dürr** received the Ph.D. degree in theoretical physics. Then, he worked ten years in a bioinformatics company developing and applying machine learning and statistical methods to all kinds of omics data. He then was a Lecturer of Statistical Data Analysis with the Zurich University of Applied Sciences. He is a Professor of Data Science with the Konstanz University of Applied Sciences. He is currently working on deep learning mainly from a probabilistic perspective.