Probabilistic Forecasting of Global Horizontal Irradiance Using A Deep Attention Based Model

Master's Thesis

June 2022

Bogi Berg, Frederik Brunø Lottrup, Jonathan Fjord Jønler

Supervised by **Kaixuan Chen**

Computer Science
Department of Computer Science
Aalborg University

AALBORG
UNIVERSITY

STUDENT REPORT

RESUMÉ

I løbet af det seneste årti er interessen for solenergi som vedvarende energikilde vokset i takt med det øgede fokus på den globale opvarmning [1]. Selvom dette er en positiv udvikling er der udfordringer forbundet med at integrere solenergi i el-netværket, da el-netværket konstant skal være i balance. Dette balancekrav afhænger af præcise prognoser af produktion og forbrug. Modsat kraftværker der anvender fossile brændstoffer, hvis produktion er forholdsvis let at forudsige, er prognosering af energiproduktionen fra solceller en svær opgave, da produktionen afhænger af mange og komplekse forhold og variable [2]. Dette er forhold og variable så som vejret, solens placering på himlen og jordens rotation. Selvom der findes en komplet og veldefineret model for solens placering og jordens rotation er vejret stadig meget svært at forudsige, selv med komplekse fysiske modeller og enorme beregningsresourcer.

Derfor har prognosering af solenergi som forskningsfelt fået meget opmærksomhed i løbet af de seneste år, hvilket har resulteret i mange forskellige forslag til hvordan man kan opnå en præcis prognose. Forudsigelse af den direkte produktion fra solceller kræver data fra eksisterende solceller, men da solceller ofte er ‘bag-måleren’ er denne data ikke lettilgængelig, fordi den eksakte produktion ikke bliver registreret. Men selv hvis at alle ejere af solceller var kendt måtte deres data ikke frigives på grund af persondataloven [3]. Derfor er forudsigelsen af solens stråling i et givet område et alternativ til forudsigelsen af den eksakte produktion. Denne enhed kaldes GHI. Da solens stråling er inkluderet i vejrmålinger, som er vidt tilgængeligt og direkte korreleret med produktionen fra en solcelle, er det muligt at anvende en prognose af solstråling for at estimere produktionen af solenergi i et specifikt område [4].

Dybe maskinlæringsteknikker bliver ofte anvendt til lignende forudsigelsesproblemer, da disse teknikker formår at lære komplekse mønstre i dataen, men en udfordring med dybe modeller er, at de ofte er ’black-box’, og derfor kan forudsigelser fra dybe modeller være svære at forklare. Denne udfordring gør, at beslutningstagere ikke kan træffe beslutninger på baggrund af disse forudsigelser, da baggrunden for beslutningerne ofte skal kunne forklares og dokumenteres [5, 6]. Denne artikel præsenterer Probabilistic Solar Irradiance Transformer (ProSIT) som er en ny end-to-end maskinlæringsarkitektur der producerer multihorizon probabilistiske forudsigelser af GHI med mulighed for øget forklarlighed. For at lære både kort- og langtids afhængigeder på tværs af input-sekvensen gør ProSIT brug af flere komplekse komponenter for at indkode det højdimensionelle feature space, såsom Bi-Directional Recurrent Neural Networks [7] der opfanger lokale temporale relationer i input sekvensen, Temporal Multi-Head Self-Attention Layers [8] som opfanger temporale afhængigheder på tværs af tidspunkter i en sekvens, hvilket giver modellen mulighed for at fokusere på flere forskellige dele af en sekvens, og Temporal Convolutional Layers [9] som afkoder modellens komplekse repræsentation, og til sidst produceres en multi-horisont kvantil prognose.

Temporal Multi-Head Self-Attention giver muligheden for at få et indblik i modellens forudsigelse, da attention mekanismen opprioriterer bestemte tidspunkter i en sekvens som den mener er af stor betydning for at kunne forudsige de næste timer frem. Disse opprioriteringer er mulige er udtrække af modellen og dermed give et visuelt indblik i modellens fokus [8]. Derudover består ProSIT af Residual Connections [10] og Gating Mechanisms [11] der er med til ad-hoc at formindske brugen af komplekse funktioner, hvor der ikke er brug for dem. Gating mekanismer og Residual Connections ‘analyserer’ produktet af en komponent og ‘beslutter’ sig for at bruge det non-lineære produkt eller reducere produktet til en lineær funktion. Det tilfører en fleksibilitet til ProSIT som reducerer risikoen for overfitting og dermed forbedrer modellens anvendelse i praksis.

Artiklen præsenterer en række eksperimenter hvor der sammenlignes mod andre state-of-the-art maskinlæringsmodeller, samt der er testet på forskellige datasæt for at demonstrere at modellen kan bruges i forskellige geografiske kontekster. ProSIT er en letvægts model sammenlignet med forhenværende state-of-the-art modeller, men opnår stadig bedre resultater testet på vejrdata fra den virkelige verden.

Probabilistic Forecasting of Global Horizontal Irradiance Using A Deep Attention Based Model

Bogi Berg
bber16@student.aau.dk

Frederik Brunø Lottrup
fbruna17@student.aau.dk

Jonathan Fjord Jønler
jjanle17@student.aau.dk

***Abstract*—The increased interest in solar energy as a renewable energy source brings new challenges to the seamless operation of the power grid due to the inherent intermittent availability. Therefore, high precision forecasts are needed to successfully integrate the growing capacity into the grid. However, forecasting the yield from solar panels is difficult due to limited data availability. Instead, GHI is forecasted. Several deep learning approaches have been proposed throughout the years, however deep learning is often considered as black-box and therefore disregarded by decision makers. In this paper we propose the Probabilistic Solar Irradiance Transformer (ProSIT), a novel end-to-end deep learning architecture for interpretable probabilistic multi-horizon forecasting of GHI. To learn both long and short-term temporal dependencies across the entire input sequence ProSIT uses several complex components to encode the high-dimensional feature space such as bi-directional recurrent networks, temporal multi-head self attention layers, and temporal convolutional layers. ProSIT also features residual connections and gating mechanisms to suppress superfluous components ad hoc. We conduct several experiments benchmarking the performance of the model, and demonstrate that ProSIT achieves state-of-the-art performance on two real-world datasets.**

***Index Terms*—Deep Learning, Solar irradiance, Multi-Horizon Forecasting, Time Series, Attention mechanisms.**

I. Introduction

In recent years, there has been an increasing interest in solar energy as a renewable and sustainable energy source due to the decrease in cost of renewable energy technologies and the growing concern about climate change [1]. However, despite being seen as the one of the most promising alternatives to fossil fuels by the industrial and scientific community, solar energy brings several challenges to the reliable and stable integration of solar energy into the power grids, due the intermittent availability of production [2]. Nevertheless, according to a report from the International Energy Agency, this does not stall the expansion of installed solar power capacity, as the global solar energy capacity has grown from 32TWh to 821TWh during the last decade, and is estimated to reach almost 7000TWh in 2030 [12]. This poses a challenge for the grid operators, as in order to ensure seamless operation of the electrical power grid, it is necessary to maintain a precise balance between the demand and supply of power, and therefore, power plant authorities use forecasts of demand and supply to decide how much power to produce.

The intermittent availability of solar energy does not only affect the grid, as the energy market functions on the basis of competitive bidding, which implies that the larger forecast error leads to a greater expenditure on energy [13]. Consequently, the unpredictability of solar energy availability can be an obstacle to its wider adoption by the energy industry and therefore its penetration in the market. Forecasting solar energy availability with high precision would allow energy providers to effectively integrate solar energy into the grid and allow energy trading companies to hedge their positions in the energy market [14]. Although the solar energy companies have the necessary historical photovoltaic (PV) data required to make predictions about energy production, they cannot share this information publicly because of data privacy policies [3].

Table I: The different factors that impact GHI [4, 15]

Variability type	Variabilities
Astronomical/Angular	the earths tilt, the earths orbit, the suns zenith, the suns azimuth
Metrological	Cloud opacity, albedo, temperature, wind speed, wind direction, relative humidity, surface pressure, precipitable water, dew point
Seasonal	Winter, Fall, Summer, Autumn

Therefore, global horizontal irradiance (GHI) is often used for prediction instead. Global horizontal irradiance is a measure of the amount of solar radiation that falls on a given surface area and is a key parameter for forecasting solar energy [4]. However, forecasting GHI is a difficult task due to the many sources of variability listed in table I.

These sources of variability can be difficult to predict and model, especially in the long-term and especially using traditional statistical methods such Auto-Regressive (AR) models. AR models are types of linear regression models that estimate the future value of a variable as a function of its past values [16, 17]. These models are typically used for forecasting financial time series data. However, they are not well suited for forecasting solar irradiance data because solar irradiance is a nonlinear function of time [18].

The use of machine learning methods for solar irradiance forecasting has steadily increased along with the rise in popularity of artificial intelligence. These methods involve learning from data patterns, modelling parameters, and building predictive models. In recent years, various different machine learning methods such as Support Vector Machines (SVM), Random Forests, XGBoost etc. have been applied for solar irradiance forecasting [19, 20, 21]. However, the aforementioned methods are so called 'shallow' models, meaning that they are generally limited in that they can suffer from over-fitting, are computationally expensive, and have low performance when handling complex and high dimensional data [22, 23]. We refer to [24] for a comprehensive review of machine learning methods applied to the problem of forecasting GHI.

Deep learning methods have been shown to be effective in overcoming the limitations of shallow machine learning models. Specifically, deep learning models are able to handle large data sets without over-fitting and do not require the same amount of extensive feature engineering as shallow models [25]. The aforementioned properties may attribute the reason for the extensive attention deep learning models have received in research the past decade. Consequently, deep learning approaches have also been applied to solar irradiance forecasting. However, contrary to shallow models and statistical models, deep learning models are generally considered to lack interpretability [5]. This means that even if a deep learning model is able to accurately classify or predict data, it may be difficult to understand how or why the model arrived at its conclusions. This lack of interpretability can be a problem when trying to use deep learning models to make decisions about complex real-world problems, as it may be difficult to understand why the model is producing certain output [6]. If a remedy to the interpretability issue was found, decision-makers could incorporate the output of deep learning models in risk-management and automation. Since GHI is used directly in the calculation of the effect of PV panels [4], an explanation of the model output would make the model suitable for providing upper and lower bounds for the power production from PV panels for a certain area.

Several approaches to alleviate this lack of interpretability have been proposed the past few years. The concept of attention has been shown to help increase interpretability of deep learning models [26]. Conceptually, attention mechanisms function similar to how human attention functions, meaning that models that enjoy attention mechanisms are able to focus on the most important parts of the input data [26]. This allows the model to better learn the relationships between the input data and the output. Furthermore, the weights tuned by said model can be plotted, allowing for visual interpretation of the output [27]. Most existing deep learning approaches for time series forecasting output point forecasts, which makes them hard to use as critical decision tools, since point forecasts offer no insight regarding the uncertainty of the prediction. On the other hand, probabilistic forecasts provide to some extent decision makers with an estimation of the forecast uncertainty [14].

In this paper, we introduce the Probabilistic Solar Irradiance Transformer (ProSIT), a novel end-to-end attention based deep learning architecture for multi-horizon probabilistic time series forecasting of GHI. Concretely, the contributions of this paper are the following:

1) We propose an attention based deep learning model inspired by the previous endeavours of [8] of producing multi-horizon forecasts.
2) We adopt the framework by [9] making our proposed model capable of producing probabilistic forecasts.
3) We conduct an ablation study demonstrating the performance gains from the individual components of the architecture, and we benchmark the scalability and computational resource requirements against state-of-the-art approaches. We evaluate the model performance of ProSIT by benchmarking against several baselines and a range of state-of-the-art approaches on two real-world historical solar irradiance data sets showing that our methods outperforms all compared methods.

Therefore, this paper is organized in the following manner: Section 2 provides a brief overview of related work on previous approaches for forecasting GHI. The underlying theoretical preliminaries for the modules constituting the architecture is described in section 3, and in section 4 an overview of the architecture is provided. In section 5, the setup of the experiments is presented, and section 6 presents the results. The results are discussed in section 7 and section 8 concludes this paper while providing future research directions.

II. Related Work

A. Applications of deep learning to solar irradiance forecasting

As stated in the previous section, deep learning methods for time series forecasting have gained a lot of attention the last few decades. One of the reasons for this could be the emergence of recurrent neural networks (RNN) from Rumelhart, Hinton, and Williams work in 1986 [28] inspiring further research leading to the invention of Long Short-Term Memory Networks (LSTM) by Hochreiter and Schmidhuber in 1997 [29]. Since then, LSTM networks have been applied to numerous time series forecasting tasks while showing prosperous results. Amongst those tasks, solar irradiance forecasting is no exception. Obiora, Ali, and Hasan applied LSTM networks to forecast the hourly solar irradiance of Johannesburg City using historical observed GHI values and meteorological variables such as temperature, sunshine duration, and relative humidity. The authors benchmark the performance of the LSTM against an SVM showing that the LSTM outperforms the SVM by 3,2% when evaluated using normalized root mean square error (nRMSE) [30]. Alharbi and Csala applied a bi-directional LSTM (BiLSTM) to solar irradiance forecasting in Tabuk City [31]. The BiLSTM architecture enjoys the capability to process historical data in a forward and backward direction, at the cost of computational complexity [7]. The authors demonstrate that the use of bi-directional RNNs exhibits promising performance when exogenous meteorological variables are introduced. Since the emergence of the transformer architecture for natural language processing tasks by Vaswani et al.[32], attention mechanisms have been applied to time series forecasting as well. Dairi, Harrou, and Sun extends the idea of utilizing a BiLSTM by adding attention to their model architecture and in doing so, the authors demonstrate that this extension is superior in performance compared to regular BiLSTMs [33]. Despite the prosperous results demonstrated by the above mentioned research efforts, the use of RNNs, whether it be bi-directional or not, limit the models capability to capture global temporal dependencies in the input data set. ProSIT alleviates this issue by incorporating a multi-head attention mechanism and a temporal convolutional network module to encode global and long term dependencies [34]. Furthermore, ProSIT utilizes several gates and residual connections to dynamically control the processing caused by each module of the model.

B. Probabilistic Time-Series Forecasting

The vast majority of the above mentioned approaches produce a point forecast, which, as stated in the previous section, does not provide decision makers with any information regarding the uncertainty of the model output. However, in the last few years, the focus has shifted towards probabilistic time series forecasting. There exists two main strategies within the area of probabilistic forecasting namely the parametric and the non-parametric. The parametric strategy for probabilistic forecasting of stochastic processes relies on the assumption that the process has a certain inherent structure that can be described using a small number of parameters, for instance the mean and standard deviation of a Gaussian distribution [35]. On the other hand, the non-parametric strategy makes no assumption regarding structure of the stochastic process, thus, a popular approach is to forecast a number of samples and model the discrete quantiles [9].

Salinas, Flunkert, and Gasthaus introduce DeepAR, a deep recurrent architecture that provides capabilities for probabilistic forecasting in an auto-regressive manner [36]. By adopting the parametric strategy the authors demonstrate that the model is able to produce forecasts estimating the parameters of probability distributions allowing for sampling in a Monte Carlo manner while remaining computationally viable. The parameters of the estimated distributions are tuned by maximizing likelihood and teacher forcing during training [36]. [37] also applies an RNN as a time series regressor, however their work takes on the non-parametric objective of estimating the quantiles of the models output by utilizing an iterative forking strategy during training. According to the authors, this avoids error accumulation commonly seen in auto-regressive methods[37]. [8] expands on the capabilities of the model aforementioned model [37]. [9] propose a framework capable of producing both of the aforementioned approaches, that is, estimating either a parametric or non-parametric probability density estimation [9]. Similar to [37] this is done in a non-auto-regressive manner, by directly forecasting the joint distribution of future observations [9]. ProSIT builds on the work of Chen et al.[9] and is therefore able to estimate both parametric and non parametric probability densities, while employing the iterative multi-horizon strategy similar to [37], however in this paper we focus primarily on demonstrating the performance of the non-parametric quantile forecasts.

III. Preliminaries

A. Multi-Horizon Forecasting

The general probabilistic multi-horizon forecasting in discrete time can be described as follows: given a length $M+1$ time series $y_{t-M:t}$ containing historical observations, we denote the future time series $y_{t:t+\Omega}$ where $\Omega \in \mathbb{Z}^+$ is the length of the forecasting horizon. The objective is to model the conditional distribution of the future time series $\mathbb{P}(y_{t+1:t+\Omega}|y_{t-M:t})$.

This conditional distribution can be written as a factorization of the joint conditional probability as such:

$$\mathbb{P}(y_{t+1:t+\Omega}|y_{t-M:t}) = \prod_{\omega=1}^{\Omega} p(y_{t+\omega}|y_{t-M:t+\omega-1}) \quad (1)$$

Where $\omega \in \{1, 2, ..., \Omega\}$ denotes the forecasting horizon. The above formula implies that every future time step is conditionally dependant on the observations at all previous timestamps. This strategy is known as the auto-regressive strategy and is used by generative models [9]. In practical scenarios, to avoid error accumulation [9], we directly forecast the joint conditional distribution:

$$\mathbb{P}(y_{t+1:t+\Omega}|y_{1:t}) = \prod_{\omega=1}^{\Omega} p(y_{t+\omega}|y_{t-M:t}) \quad (2)$$

Often, the target time series y has inherent patterns and dependencies affected by covariates $X^{(i)}_{t-M:t}$ where $i = 1, ..., N$, N being the number of covariates. We denote $\gamma_{t-M:t} = (y_{t-M:t}, X^{1:N}_{t-M:t})$ the concatenation of the past observations and the past covariates, hence the modelling objective of the conditional distribution becomes:

$$\mathbb{P}(y_{t+1:t+\Omega}|y_{t-M:t}) = \prod_{\omega=1}^{\Omega} p(y_{t+\omega}|\gamma_{t-M:t}) \quad (3)$$

The task at hand is now to output the multi-horizon quantile forecasts $\hat{\mathbf{y}}^q_{t+1:t+\Omega}$ for quantile $q \in Q$, where $Q \in [0, 1]$ using a model $\mathcal{F}$ with parameters of Θ such that

$$\hat{\mathbf{y}}^q_{t+1:t+\Omega} = \mathcal{F}_\Theta(\gamma_{t-M:t}, q) = \mathbb{P}(\mathbf{y}_{t+1:t+\Omega} \leq \hat{\mathbf{y}}^q_{t+1:t+\Omega}|\gamma_{t-M:t}) \quad (4)$$

Which is done by means of quantile regression described in the next section.

B. Quantile Regression

ProSIT obtains forecasts by means of quantile regression introduced by Koenker and Bassett in 1978 [38]. This means that the model is trained using quantile loss defined as:

$$L_q(y, \hat{y}^q) = q(y - \hat{y}^q)_+ + (1 - q)(\hat{y}^q - y)_+ \quad (5)$$

where y is the observed target, $\hat{y}^q$ is the prediction for a specific quantile q, hence $q \in \{x \,|\, 0 \leq x \leq 1\}$. Furthermore, $(y)_+ = \max(0, y)$. Let $Q = \{q_1, q_2, ..., q_k\}$ be a set of k quantile levels, the corresponding k forecasts are obtained by minimizing the total quantile loss:

$$L_Q = \sum_{j=1}^{k} L_{q_j}(y, \hat{y}^{q_j}) \quad (6)$$

C. Bi-directional Recurrent Neural Networks

1) Gated Recurrent Units: As stated previously when reviewing existing literature, Rumelhart, Hinton, and Williams [28] paved the way for Recurrent Neural Networks allowing for deep sequence modelling. This framework was later extended by [29] with their architecture the Long Short-Term Memory network (LSTM), and a decade later in 2014 the Gated Recurrent Unit emerged (GRU) from the works of Cho et al.[39]. The introduction of gates in the RNN architecture alleviated the issue of vanishing gradients, a phenomenon occurring when contribution of past information in a given sequence tends towards very small numerical values causing the network to 'forget' what it has seen [29, 39]. Figure 1 shows an example of a GRU. The generic GRU architecture features two gates, the update gate and the reset gate, which are illustrated on the figure with coloured regions. These gates allow the model to determine how much of the past information needs to be passed along to the future and how much of the past information to forget [40]. The GRU model can be expressed as a set of equations that demonstrates the numerical behaviour of the update and reset gate as well as the auto-regressive way of handling input:

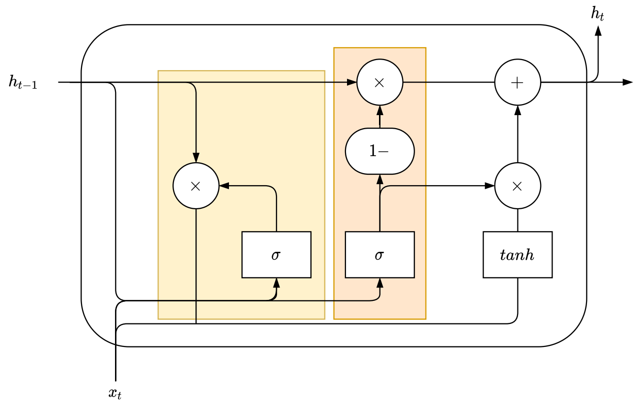

Figure 1: Inside a gated recurrent unit cell

$$\mathbf{z_t} = \sigma(\mathbf{W}_z\mathbf{x}_t + \mathbf{U}_z\mathbf{h}_{t-1} + \mathbf{b}_z) \tag{7}$$

$$\mathbf{r_t} = \sigma(\mathbf{W}_r\mathbf{x}_t + \mathbf{U}_r\mathbf{h}_{t-1} + \mathbf{b}_r) \tag{8}$$

$$\hat{\mathbf{h}}_t = \tanh(\mathbf{W}_h\mathbf{x}_t + \mathbf{U}_h(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}_h) \tag{9}$$

$$\mathbf{h}_t = \mathbf{z}_t \odot \hat{\mathbf{h}}_t + (1 - \mathbf{z_t}) \odot \mathbf{h}_{t-1} \tag{10}$$

where $\mathbf{x}_t$ and $\mathbf{h}_t$ denote the input and output vectors respectively, $\hat{\mathbf{h}}_t$ denotes the candidate state vector that controls the influence level of the previous hidden state. Moreover, the update and reset gates are denoted with vectors $\mathbf{z}_t$ and $\mathbf{r}_t$ respectively. The aforementioned gates and layers are parameterized by the weight matrices $\mathbf{U}_{(.)}$, $\mathbf{W}_{(.)}$ and bias vector $\mathbf{b}_{(.)}$. The activation functions of the architecture are the logistic sigmoid function denoted by σ and the hyperbolic tangent [40]. Henceforth, the above equations defining the GRU architecture will be abbreviated as $GRU(\mathbf{x}_t, \mathbf{h}_{t-1})$.

2) Bi-Directional Recurrent Neural Networks: The bi-directional RNN is a concept invented by Schuster and Paliwal which dates back to the conception of the LSTM [7]. The bi-directional property can be applied to any RNN architecture, including gated RNNs, by stacking an additional RNN that processes the input in a backwards manner, as illustrated on figure 2.

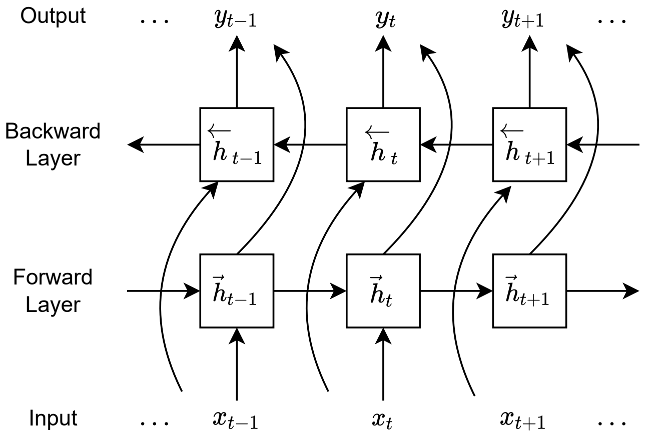

Figure 2: A Bi-Directional Recurrent Neural Network[7].

Applying the bi-directionality property to a GRU results in the following expressions [41]:

$$\overleftarrow{\mathbf{h}}_t = GRU(\mathbf{x}_t, \overleftarrow{\mathbf{h}}_{t-1}) \tag{11}$$

$$\overrightarrow{\mathbf{h}}_t = GRU(\mathbf{x}_t, \overrightarrow{\mathbf{h}}_{t-1}) \tag{12}$$

$$\mathbf{h}_t = \overrightarrow{\mathbf{h}}_t + \overleftarrow{\mathbf{h}}_t \tag{13}$$

D. Residual Connections & Gating Mechanism

1) Residual Connections: As initially stated in the first section of this paper, deep neural networks (DNN) hold an advantage over shallow networks due to their capability to learn complex feature representations automatically and approximate functions of an arbitrary number of dimensions. However, it has been shown that as the depth of deep learning models increases, their performance in terms of training accuracy starts degrading [42, 43]. Note that we use the terminology 'weight layer' as a general term for a any kind of layer, whether it be linear or convolutional layer. In an effort to alleviate this degradation, He et al. propose the deep residual framework, illustrated on figure 3 [10].

As discussed in [10], one of the main advantages of residual neural networks compared to plain neural networks, is that instead of fitting the layers in a network a desired mapping $\mathcal{H}(x)$, the layers of the network are trained to fit a residual mapping $\mathcal{G}(x) = \mathcal{H}(x) - x$ such that the desired mapping is recast as $\mathcal{G}(x) + x$ [10].

2) Gating Mechanisms: It has been shown that gates in DNNs may improve model performance, for instance in the GRU architecture, by controlling the information propagation throughout the model [29, 40]. However, the concept of gating is not exclusive to RNNs as shown in the works by Dauphin

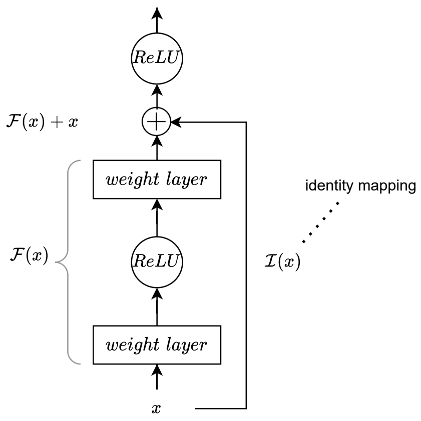

Figure 3: He et al.'s residual building block

et al. [11] in which they propose the Gated Linear Unit (GLU) and show an increase in model performance. Dauphin et al.'s describe the GLU as a simplified version of [44]'s work, and express the GLU using the following equation:

$$\mathrm{GLU}(\mathbf{X}) = \sigma(\mathbf{W}\,\mathbf{X}+\mathbf{b}) \odot (\mathbf{X}\,\mathbf{V}+\mathbf{c}) \tag{14}$$

where $\mathbf{X}$ is the input, $\mathbf{W}$, $\mathbf{V}$ are weight matrices, $\mathbf{b}, \mathbf{c}$ are bias vectors, and $\odot$ denotes the element-wise Hadamad Product [11]. Figure 4 shows a graphical representation of the GLU. The gating property of the GLU comes from the non-linearity of the sigmoid function, which squishes the input into the range between 0 and 1, hence if the input contributes positively towards the optimization goal, the GLU approaches the behaviour of an identity function. Conversely, if the input contributes negatively, the output of the GLU approaches zero, essentially cancelling out the input.

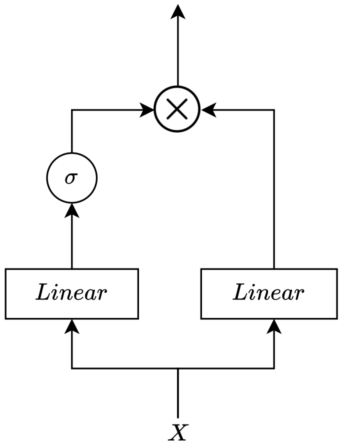

Figure 4: The Gated Linear Unit

3) Gated Residual Network: Lim et al. adopts these aforementioned concepts of the residual block and the GLU and incorporates them into their proposed Gated Residual Network (GRN) [8], as depicted on figure 5. The graphical illustration can be expressed mathematically as:

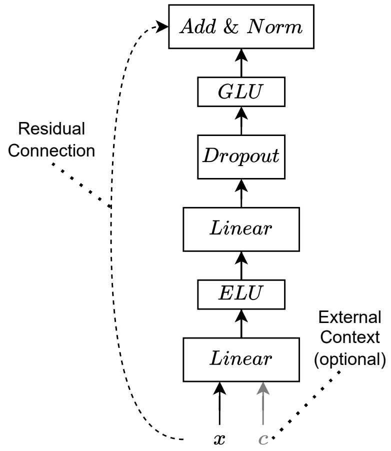

Figure 5: The Gated Residual Network, as described in [8]

$$\mathrm{GRN}(\mathbf{x}, \mathbf{c}) = \mathrm{LayerNorm}(\mathbf{x} + \mathrm{GLU}(\eta_1)) \tag{15}$$

$$\eta_1 = \mathbf{W}_1 \eta_2 + \mathbf{b}_1 \tag{16}$$

$$\eta_2 = \mathrm{ELU}(\mathbf{W}_2\mathbf{x} + \mathbf{W}_3\mathbf{c} + \mathbf{b}_2) \tag{17}$$

$$\mathrm{ELU}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha(e^x - 1) & \text{if } x \leq 0 \end{cases} \tag{18}$$

Where $\mathbf{x}$ is the input to the GRN, $\mathbf{c}$ is the optional external context vector, $\mathbf{W}_{(.)}$ and $\mathbf{b}_{(.)}$ are weights and biases, LayerNorm is the layer normalization introduced by [45], $\mathrm{ELU}(x)$ is the Exponential Linear Unit as introduced by [46], and η_1, η_2 are intermediate fully connected layers in accordance with the description of [8]. For clarity, equations 15-18 are generalizations of the expressions covered in [8].

E. Multi-Head Self-Attention

The self-attention mechanism is a key ingredient in the transformer architecture that has shown great success in various natural language processing tasks [47, 32, 48]. Multi-head self-attention allows the model to jointly attend to information from different representation subspaces at different positions [27]. This is beneficial for time series forecasting as it allows the model to learn representations that are more robust to changes in the input over time [49, 33].

Concretely, attention mechanisms work by scaling the input values $\mathbf{V} \in \mathbb{R}^{N \times d_V}$ in accordance with relationships between keys $\mathbf{K} \in \mathbb{R}^{N \times d_{attn}}$ and queries $\mathbf{Q} \in \mathbb{R}^{N \times d_{attn}}$ as such [32]:

$$\mathrm{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathrm{A}(\mathbf{Q}, \mathbf{K})\mathbf{V} \tag{19}$$

Where A(.) is the scaled dot-product, which serves as a normalization function [32]:

$$\mathrm{A}(\mathbf{Q}, \mathbf{K}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_{attn}}}\right) \tag{20}$$

By extending the above expression to the notion of Multi-head attention we obtain [8]:

$$\text{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) = [\mathbf{H}_1, \mathbf{H}_2, ..., \mathbf{H}_{m_H}]\mathbf{W}_H \tag{21}$$

$$\mathbf{H}_h = \text{Attention}\Big(\mathbf{Q}\mathbf{W}_Q^{(h)}, \mathbf{K}\mathbf{W}_K^{(h)}, \mathbf{V}\mathbf{W}_V^{(h)}\Big) \tag{22}$$

where the head specific weights, denoted by $\mathbf{W}_Q^{(h)}, \mathbf{W}_K^{(h)} \in \mathbb{R}^{N \times d_{attn}}$ and $\mathbf{W}_V^{(h)} \in \mathbb{R}^{N \times d_V}$, for the query, key and values respectively. Furthermore, the linear combination of the concatenated outputs are denoted by $\mathbf{W}_H$, where H is the number of heads.

Lim et al. further expands the interpretability by employing an additive aggregating strategy to the weights of the heads, which results in [8]:

$$\text{InterpretableMultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \tilde{\mathbf{H}}\mathbf{W}_H \tag{23}$$

where $\tilde{\mathbf{H}}$ is defined by

$$\tilde{\mathbf{H}} = \tilde{A}(\mathbf{Q},\mathbf{K})\mathbf{V}\mathbf{W}_V \tag{24}$$

$$= \Big\{\frac{1}{H}\sum_{h=1}^{m_H} A\Big(\mathbf{Q}\mathbf{W}_Q^{(h)}, \mathbf{K}\mathbf{W}_K^{(h)}\Big)\Big\}\mathbf{V}\mathbf{W}_V \tag{25}$$

$$= \frac{1}{H}\sum_{h=1}^{m_H} \text{Attention}(\mathbf{Q}\mathbf{W}_Q^{(h)}, \mathbf{K}\mathbf{W}_K^{(h)}, \mathbf{V}\mathbf{W}_V) \tag{26}$$

and the matrix $\mathbf{W}_H \in \mathbb{R}^{d_{attn} \times N}$ is used for the final linear projection [8].

F. Temporal Convolutional Networks

The temporal convolutional network with dilated causal convolutions is a deep learning architecture that has been proposed by Lea et al.[50], which is evaluated in [51] for the task of sequence modelling [51]. The network consists of a number of layers of residual blocks, shown on figure 7, featuring dilated causal convolutions (DCC) [50]. An illustration of a DCC layer is shown on figure 6. DCCs are shown to be more effective than traditional convolutional networks for modelling sequences [50].

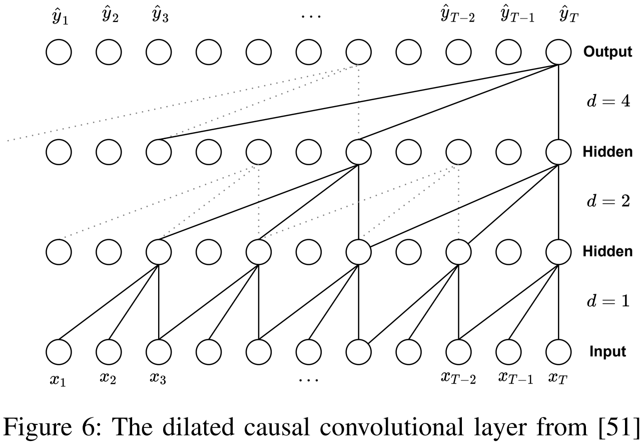

Figure 6: The dilated causal convolutional layer from [51].

Formally, A DCC operation can be expressed as such:

$$F(s) = (\mathbf{x} *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot \mathbf{x}_{s-d \cdot i} \tag{27}$$

where $f : \{0, ..., k-1\} \mapsto \mathbb{R}$ is a filter and $\mathbf{x} \in \mathbb{R}^l$ denotes the one dimensional input of length l, k denotes the kernel size of the convolutions, and d denotes the dilation size [50].

The TCN architecture is capable of processing inputs of arbitrary length, however, the number of layers is bounded by the length of the input, as shown in [51]. Concretely, let l be the length of the input for a TCN with dilation base b and kernel size k, the receptive field needed to cover the entirety of l must be at least

$$1 + (k-1)\frac{b^n - 1}{b - 1} \geq l \tag{28}$$

We solve for n to obtain minimum number of layers and hereby the minimal depth for the network as such:

$$n = \Big\lceil log_b\Big(\frac{(l-1)(b-1)}{k-1} + 1\Big)\Big\rceil \tag{29}$$

Consequently, this results in very deep networks for very long inputs, which may be subject to the aforementioned degradation issue [50]. Therefore, the TCN architecture usually features residual connections [51].

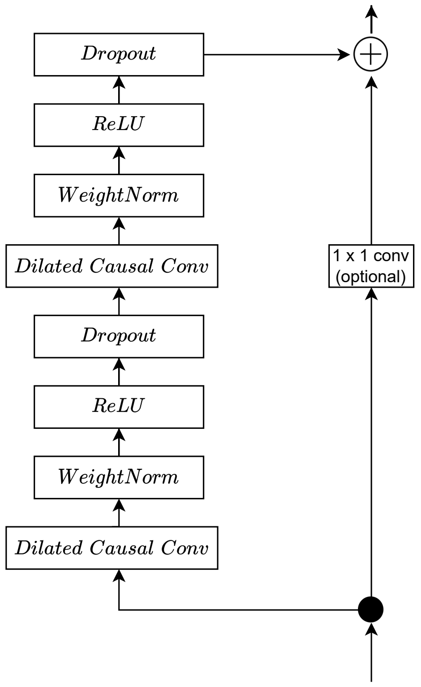

Figure 7: A Residual block of the TCN architecture [51, 50].

This concludes the theoretical background review. In the next section we will address how each of these aforementioned components and methods are used in ProSIT furthermore, we will cover which properties each individual component provides to the model.

IV. Model Architecture

In this section we outline the architecture of ProSIT, formally define the functions and operations transforming the input to a probabilistic forecast, and motivate use of modules in the architecture. The complete architecture of ProSIT is depicted in figure 8. The proposed ProSIT consists of three modules: Covariate encoder, temporal transformer, and the long term dependency decoder module.

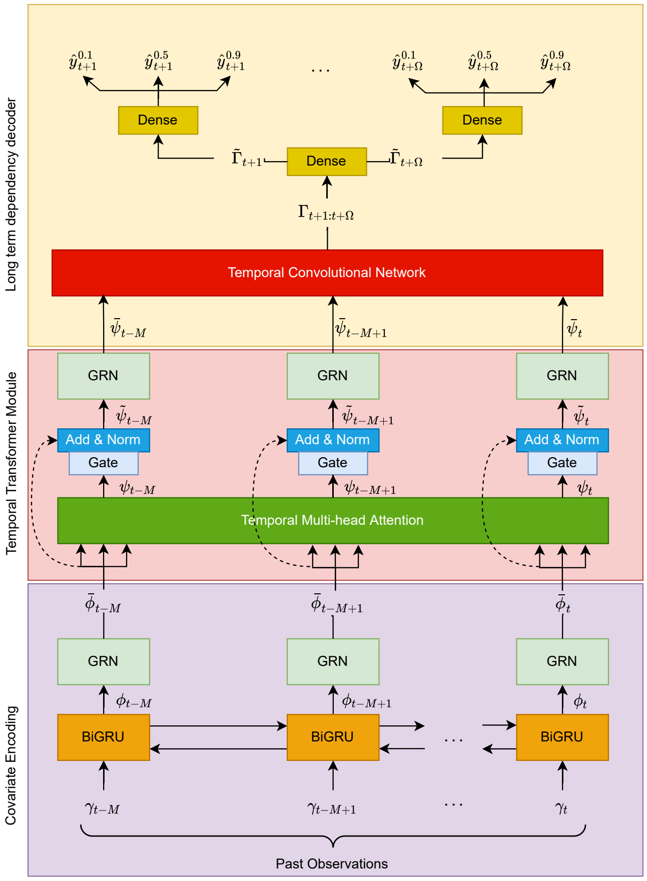

Figure 8: The architecture overview of ProSIT

A. Covariate Encoder

The purpose of the covariate encoder is to encode the relationships between target $\mathbf{y}_{t:t+\Omega}$ and the past covariates $\mathbf{X}_{1:t}$. As such the encoder consists of a BiGRU which gives the model the capability to capture temporal dependencies in both directions. The reason we chose a GRU architecture for the encoder instead of an LSTM is that it has been shown that a GRU is less computationally expensive to train, as the GRU architecture features only two gates compared to the three in the LSTM. The missing gate is made up for in the GRN that follows directly after the encoder. A benefit of moving the gate outside of the RNN the ability to parallel process the output, which is not possible if the gate was situated within the RNN. We chose the bi-directional architecture as short-term temporal patterns such as changes in cloud cover are desirable to capture for solar irradiance forecasting. Capturing future patterns in the input could also have been achieved through a future covariate encoder, as featured in [8], however, this comes at an increased computational cost.

We consider a lookback window into the past observations of length M as the input for the model, hence the processing of the input by the encoder can formally be expressed by letting $\gamma t - M : t = (\mathbf{y}_{t-M:t}, \mathbf{X}_{t-M:t}) \in \mathbb{R}^{N+1 \times M}$ denote the N past covariates and past observed GHI values as the input for the encoder module with hidden size d_{model}. The BiGRU encodes the input by applying the following processing:

$$\phi_{t-M:t} = \text{BiGRU}(\gamma_{t-M:t}, \mathbf{h}_0), \tag{30}$$

where $\mathbf{h}_0$ denotes the initial state of the BiGRU and $\phi_{t-M:t} \in \mathbb{R}^{d_{model} \times M}$ denotes the output of the BiGRU and represents the temporally encoded input $\gamma t - M : t$. Please note that we use the shorthand notation $(\cdot)_{t-M:t}$ to denote a whole sequence, where in practice, the BiGRU processes the input sequence one step at the time.

There might be cases where the complex non-linear processing of the input would be superfluous to the model performance, for instance during the night hours where the GHI is constant zero. To address this, ProSIT enjoys the capability to disregard the contribution from the various preceding components of the architecture, a capability provided by the GRN layers placed between the modules, as seen on figure 8. Since predicting zeros has been shown to be difficult for deep models [52], disregarding or diminishing the contribution of the complex components at certain time steps allows the model to accurately predict the absence of GHI during the night hours. Formally, this can be expressed by letting $\bar{\phi}_{t-M:t} \in \mathbb{R}^{d_{model} \times M}$ denote the output sequence perturbed by the GRN:

$$\bar{\phi}_{t-M:t} = \text{GRN}(\phi_{t-M:t}), \tag{31}$$

where $\bar{\phi}_{t-M:t}$ represents the information retained by the gating property of the GRN. As Lim et al. points out this property is attributed to the numerical behaviour of the ELU activation function found in the GRN. Recall eq. 17 and 18 and observe that when $\mathbf{W}_2\mathbf{x} + \mathbf{W}_3\mathbf{c} + \mathbf{b}_2 >> 0$ then ELU behaves as an identity function, and conversely when $\mathbf{W}_2\mathbf{x} + \mathbf{W}_3\mathbf{c} + \mathbf{b}_2 << 0$ the ELU emits a constant output thus acting as a linear layer [8]. An important distinction from the GRN layer proposed by Lim et al. and the GRN in ProSIT is the exclusion of external context, consequently, since ProSIT does not posses the ability to process static covariates, we omit the input $\mathbf{c}$ from the GRN [8].

B. Temporal Transformer

Inspired by the Temporal Fusion Transformer (TFT) by Lim et al., ProSIT employs a multi-head attention mechanism to capture different patterns across multiple timesteps in the input sequence that may be difficult for the BiGRU to capture. Concrete examples of patterns are the steep decline and increase in GHI from day to night and vice versa, or the subtle effects when a cloud covers the sky temporarily during the day, resulting in a sudden decrease in GHI followed by an increase. Since we adopt the attention mechanism as the TFT, ProSIT enjoys the same qualities in terms of interpretation capabilities [8].

The temporal multi-head attention layer applies the following manipulation to $\bar{\phi}_{t-M:t}$:

$$\psi_{t-M:t} = \text{InterpretableMultihead}(\bar{\phi}_{t-M:t}, \bar{\phi}_{t-M:t}, \bar{\phi}_{t-M:t}) \tag{32}$$

where $\psi_{t-M:t} \in \mathbb{R}^{d_{model} \times M}$ represents the manipulation caused by the attention weight on the retained gated information from the preceding gated input $\bar{\phi}_{t-M:t}$. ProSIT features a component skip-gate between following the temporal multi-head attention layer that skip said layer if the perturbation from the layer results in increased loss. This gate is followed by another GRN:

$$\bar{\psi}_{t-M:t} = \text{GRN}(\tilde{\psi}_{t-M:t}), \tag{33}$$

$$\tilde{\psi}_{t-M:t} = \text{LayerNorm}(\bar{\phi}_{t-M:t} + \text{GLU}(\psi_{t-M:t})), \tag{34}$$

where $\bar{\psi}_{t-M:t} \in \mathbb{R}^{d_{model} \times M}$ denotes the output from the temporal transformer module, and $\tilde{\psi}_{t-M:t} \in \mathbb{R}^{d_{model} \times M}$ denotes the gated normalized sum of $\bar{\phi}_{t-M:t}$ and the gated output of the temporal multi-head attention layer $\psi_{t-M:t}$.

C. Long-range Dependency Decoder

The long-range dependency decoder serves the purpose of interpreting the attention weighed sequence from the temporal transformer module to capture the long-range latent correlations between the past covariates and the target. ProSIT employs a TCN to provide such a capability:

$$\zeta_{t+1:t+\Omega} = \text{TCN}(\bar{\psi}_{t-M:t}) \tag{35}$$

Where $\zeta_{t+1:t+\Omega} \in \mathbb{R}^{d_{model} \times \Omega}$ denotes the output sequence of length Ω.

We chose the TCN architecture, as it has been shown that the TCN is able to handle sequences of arbitrary length without the shortcomings of recurrent architectures such as vanishing gradients [9, 50], and since this is the last non-linear processing preceding the output of the model, relevant information from the entire length of the sequence must be retained.

ProSIT enjoys the capability of producing quantile outputs, a capability facilitated by passing $\zeta_{t+1:t+\Omega}$ through a dense layer parameterized by weight matrix $\mathbf{W}_\alpha$ and bias vector $\mathbf{b}_\alpha$, and thereafter distributing each time step from the output of the TCN through a dense layer:

$$\tilde{\zeta}_{t+1:t+\Omega} = \mathbf{W}_\alpha \zeta_{t+1:t+\Omega} + \mathbf{b}_\alpha \tag{36}$$

$$\hat{y}^q_{t+1:t+\Omega} = \mathbf{W}^q_\omega \tilde{\zeta}_{t+\omega} + \mathbf{b}^q_\omega \text{ for } \omega = \{1, ..., \Omega\}, q \in [0, 1] \tag{37}$$

where $\mathbf{W}^q_\omega, \mathbf{b}^q_\omega$, is the weight matrix and bias vector for that specific time step and quantile respectively.

D. Evaluation Metrics & Loss Functions

As mentioned in III-B the model is trained by jointly minimizing the summed quantile loss across all selected quantiles [8]:

$$\mathcal{L}(\mathcal{D}_{train}, \Theta) = \sum_{y_t \in \mathcal{D}_{train}} \sum_{q \in Q} \sum_{\omega=1}^{\Omega} \frac{L_q(y_t, \hat{y}^q_t)}{\mathcal{M}_\Omega} \tag{38}$$

where $\mathcal{D}_{train}$ denotes the domain of training data consisting of $\mathcal{M}$ samples, Θ is the weights of ProSIT, the set Q denotes the output quantiles (for the sake of comparability we use $Q = \{0.1, 0.5, 0.9\}$ like [8]).

We quantitatively evaluate the performance of our model compared to several other models using the Normalized q-risk error metric. Normalized q-Risk for a specified quantile q is for a univariate stochastic predicted timeseries given by [8]:

$$\text{q-Risk} = \frac{2 \sum_{y_t \in \mathcal{D}_{test}} \sum_{\omega=1}^{\Omega} L_q(y_t, \hat{y}^q_t)}{\sum_{y_t \in \mathcal{D}_{test}} \sum_{\omega=1}^{\Omega} |y_t|} \tag{39}$$

Where $\mathcal{D}_{test}$ denotes the domain of test samples.

In line with customary practices we also conduct benchmarks against naive persistence models, to determine the value added by the model in question. As such we use the Mean Absolute Scaled Error (MASE) [53]:

$$MASE = \frac{\frac{1}{\|\mathcal{D}_{test}\|} \sum_{y_t \in \mathcal{D}_{test}} \sum_{\omega=1}^{\Omega} |y_t - \hat{y}_t|}{\frac{1}{\|\mathcal{D}_{train}\|-1} \sum_{y_t \in \mathcal{D}'_{train}} \sum_{\omega=1}^{\Omega} |y_t - y_{t-1|}} \tag{40}$$

Where $\mathcal{D}'_{train} = y_{2:t}$ is the training dataset where the first value is omitted and $\| \cdot \|$ denotes the length of the argument. Essentially, MASE is the mean absolute error of a forecast scaled by the mean absolute error of a naive time series forecast [53]. MASE is therefore a measure of the performance of the forecast compared to the performance of a naïve forecast, i.e the forecast for time $t+1$ is always the last value. We evaluate this metric on the median quantile.

V. Experiments & Implementation Details

A. Datasets

We chose two datasets to demonstrate the applicability and generality our model. The two datasets are situated in two different locations with varying climate. We chose different climates to display the adaptability of ProSIT . The datasets used for training and experiments are provided by Solcast [54] and contains time-series of astronomical and metrological variables and GHI for two areas: Groningen in the Netherlands and Brighton in the United Kingdom, referred to as `NL-GHI` and `UK-GHI` respectively. Typically, the inland climate of `NL-GHI` is characterized by higher variance in weather compared to the coastal climate of `UK-GHI` as the ocean regulates the weather [55]. However, the coastal climate of `UK-GHI` typically means more precipitation during the summer season compared to NL-GHI [55]. The data is from 2007 to 2022 and has a granularity of one hour, accounting for 131.400 data points in total. The variables and their units are listed in table II. The observations and measurements of the variables are made at a 10m height measured from the ground up.

Table II: Dataset variables

Variable	**Unit**
Global Horizontal Irradiance (GHI)	$\frac{W}{m^2}$
Direct Normal Irradiance (DNI)	$\frac{W}{m^2}$
Direct (Beam) Horizontal Irradiance (EBH)	$\frac{W}{m^2}$
Diffuse Horizontal Irradiance (DHI)	$\frac{W}{m^2}$
Cloud Opacity	%
Albedo (daily average)	%
Solar Zenith	$^\circ$
Solar Azimuth	$^\circ$
Temperature	$^\circ C$
Wind Speed	$\frac{m}{s}$
Wind Direction	$^\circ$
Relative Humidity	%
Surface Pressure	hPa
Precipitable Water	$\frac{kg}{m^2}$
Dew Point	$^\circ C$

B. Training Procedure

The optimization process during training of the models (ProSIT and the benchmarks) was done using the ADAM optimizer with a weight decay of 0.000005 [56] and the learning rate scheduling method called Cosine Annealing with Warm Restarts [57] with a restart rate of 60 epochs. For numerical stability we employ a min-max scaling [58] on each of the individual variables. Moreover, we employ a sine/cosine transformation to cyclical features such as time, azimuth, zenith and wind direction.

1) Hyper-Parameter Search: For training and evaluation purposes we partition the two datasets into two parts, namely a training set for learning and a validation set for hyper-parameter search and performance evaluation. The grid-search strategy was chosen for identifying the best performing hyper-parameters for the model among the configurations listed below. Each configuration was run for 50 epochs and evaluated based on the performance on the median quantile, measured using the Rho-Risk metric.

- **Lookback Window** - 72, 96, 120
- **Hidden Size** - 80, 92, 128
- **Dropout Rate** - 0.1, 0.3, 0.5
- **Learning Rate** - 0.0001, 0.001, 0.01
- **Batch Size** - 64, 128, 256
- **Kernel Size** - 3, 4, 5
- **Dilation Size** - 2, 3

The configurations resulting in the best model performance for each dataset can be seen in table III.

Table III: The best performing configurations

	NL-GHI	UK-GHI
Dataset Details		
Climate	Inland	Coastal
Model Parameters		
Lookback Window	120	120
Horizon	5	5
Hidden Size	128	92
Number of Heads	4	4
Kernel Size	4	3
Dilation Size	2	2
Dropout Rate	0.3	0.3
Training Parameters		
Batch Size	128	128
Learning Rate	0.0001	0.0001
Weight Decay	0.00005	0.00005

2) Implementation Details: The data from Solcast was of the format .csv and was processed and analyzed using the framework by PyData Pandas [59]. The proposed model has been implemented in Python using the PyTorch library [60]. For training and experiments, the Python framework for time-series Darts version 0.16.1 [61] has been used. The experiments have been run on a machine running on Linux Ubuntu 20.04.4 with NVIDIA TESLA P100 GPUs and an Intel(R) Xeon(R) CPU 2.20GHz with two cores, provided for free by Kaggle.

C. Computational Cost

Figure 9 shows the benchmark of computational cost between ProSIT and the state-of-the-art model the Temporal Fusion Transformer demonstrating the 'frugality' of our model in terms of computational cost. These results clearly indicate that even though ProSIT is a complex model, the computational resources required to train the model is not extensive compared to the Temporal Fusion Transformer. Concretely, our model is over five times faster on the largest datasets.

D. Benchmarks

To reason about the performance of ProSIT we benchmark against a wide range of models. Prior to training and evaluating the models we perform a hyper-parameter search using the grid search strategy on a pre-defined fixed search space, using the same number of epochs across all models. The selected models for comparison are ARIMA [17] , GRU [39], a bidirectional LSTM [31], A Temporal Convolutional Network [9], Temporal Fusion Transformer [8]. The Bi-LSTM and GRU have been modified such that they are capable of producing quantile outputs in accordance with the framework of [9].

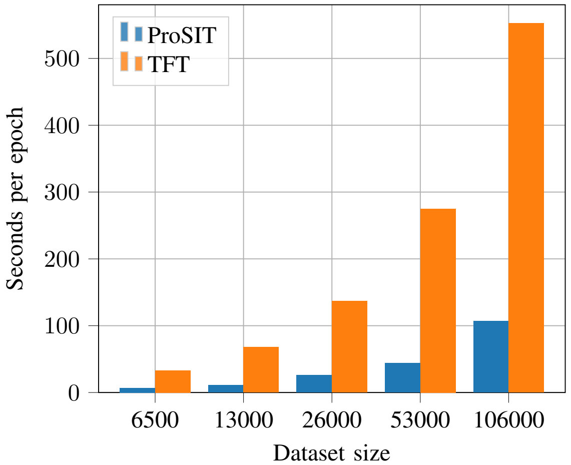

Figure 9: Comparison in run-time per epoch between the Temporal Fusion Transformer [8] and ProSIT

Table IV: Benchmarks

P50, and P90 quantile losses on two GHI datasets with varying climate, with ProSIT outperforming all competing methods across all experiments.

	ARIMA		**GRU**		**BiLSTM**	
	P50	P90	P50	P90	P50	P90
NL-GHI	9.173	21.48	4.150	5.936	5.567	9.248
UK-GHI	15.42	24.32	5.519	1.905	1.946	1.180
	TCN		**TFT**		**ProSIT**	
	P50	P90	P50	P90	P50	P90
NL-GHI	4.525	7.057	0.239	0.054	**0.050**	**0.022**
UK-GHI	4.761	7.824	0.430	0.090	**0.036**	**0.017**

Table V: MASE Benchmarks

The mean absolute scaled error benchmarks on two GHI datasets, lower values are better.

	ARIMA	**GRU**	**BiLSTM**	**TCN**	**TFT**	**Proposed**
NL-GHI	4.848	1.656	1.855	2.611	1.022	**0.846**
UK-GHI	6.838	2.994	2.017	2.465	0.950	**0.807**

VI. Results & Discussion

Table IV shows that ProSIT outperforms all of the benchmark methods by several magnitudes on both datasets described in the previous section. For median (P50) forecasts, ProSIT outperforms the TFT, which is considered as the former state-of-the-art architecture. In conjunction with figure 9 we see that not only does ProSIT outperform complex models in terms of performance, but our approach is also computationally cheaper. The same pattern emerges when inspecting the P90 error, where ProSIT also outperforms the TFT significantly. This demonstrates that ProSIT is not only a cheaper but better alternative to the TFT for probabilistic forecasting of GHI.

The results of table IV also indicate that ProSIT is robust with regard to datasets, as the model performs similar on both datasets, demonstrating that the architecture is capable of handling datasets originating from both inland and coastal climates. Moreover, table V shows the performance of ProSIT compared to the aforementioned benchmark models, once again indicating that ProSIT outperforms the competition significantly.

A. Ablation Study

As the proposed model consists of multiple components, an ablation study was conducted to examine whether the individual components are improving the performance of the model. The ablation study included three different configurations (a,b and c) of the model. Configuration **a** was the full architecture as illustrated in figure 8. In configuration **b**, the temporal multi-head attention layer and the succeeding gate and add & norm layers were removed. For configuration **c**, the bi-directional GRU layer in the covariate encoder was modified to be unidirectional, and all other components were as illustrated in figure 8. The training- and validation loss of the different configurations is depicted on figure 10 (a, b and c). The measured errors are the sum of all quantiles (0.1, 0.5 and 0.9).

The results concluded that the full architecture (configuration **a**) provided the best results with training loss and validation loss of 0.04170 and 0.04690 respectively. Configuration **c** provided the second best results with training and validation loss of 0.06 and 0.0706 respectively, while configuration **b** proved to be the least successful with training and validation loss of 0.0832 and 0.244 respectively - an increase in loss of almost 200% on the training set compared to the full model, and over 500% worse on the validation set. The loss curves of configurations **a** and **c** (figure 10 (a) and (c)) show that the model has a smooth degrading loss through the learning epochs, and both models seem to be converging around the 80th epoch. The minor bumps in the loss curves are caused by the cosine annealing with warm restarts in the learning scheduler. The distance between the training loss and the validation loss in configuration **b** is noteworthy, as this shows that without the attention mechanism, the architecture is very prone to overfitting. In conclusion, every component contributes to the model in some way. Especially the temporal multi-head component enables the model to generalize and learn well. The increase in encoding capability provided by the bidirectionalty property of the BiGRU also contributes positively towards minimal loss.

B. Quantile Outputs

As initially stated, one of the shortcomings of point forecasts is the lack of information regarding the range of potential future outcomes. ProSIT is capable of producing non-parametric probabilistic forecasts, as seen on figure 11, providing decision makers with prediction intervals in the form of quantiles. Figure 11a shows forecasts on the `NL-GHI` dataset, and 11b shows forecasts on the `UK-GHI` dataset. The figure clearly shows that the model on both datasets is quite

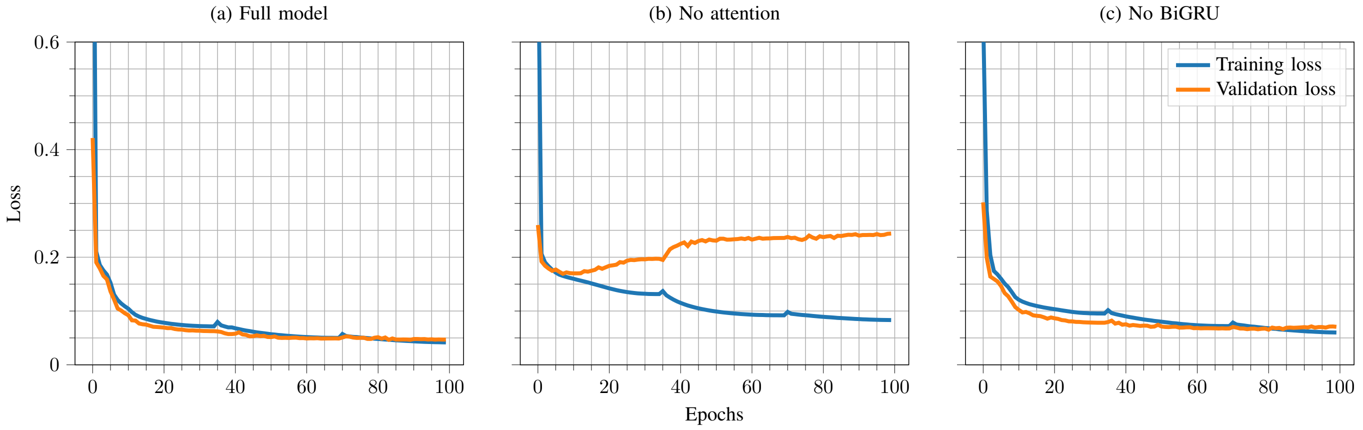

Figure 10: Ablation Study Results

certain during the night hours, which could be an indication that the model utilizes the gates of the architecture and thereby relying on a more simple representation instead of the complex non-linear processing. Furthermore, the figure shows that the model is capable of modelling both clear sky days, i.e. days where there is none or minimal clouds, and to some extend model the uncertainty that is inherent in days with mixed sky cover.

C. *Interpreting Attention*

As initially stated, deep learning approaches often lack interpretability thus precluding deep learning models from being used by decision-makers in critical business areas. Consequently, being able to interpret a models decision is of high value. The Temporal Transformer module offers the capability to obtain such insight into the underlying mechanisms for prediction, as it captures patterns in the input sequence.

Figure 12 illustrates 120 timesteps from the past and a forecast horizon of 5 timesteps into the future along with attention scores for every time step. The peaks in GHI indicate the day hours, and the valleys indicate the night hours. The forecast is in the day hours. It is noteworthy that the attention score has two peaks, one during the day hours of the most recent day, and one during the day hours from four days ago. This indicates that the GHI peaks have an influence on the attention, however, it is obvious that some of the historic timesteps have no influence on the attention scores. Contrary, if the model is to forecast GHI during the night hours, the peaks in attention scores also happen during the night hours of the input.

VII. Conclusion

We present ProSIT, a novel end-to-end attention based deep learning architecture for multi-horizon probabilistic time series forecasting of GHI. To handle long and short term temporal dependencies and relationships ProSIT consists of three modules: (1) The covariate encoder that captures local temporal relationships between past observed covariates and the target variable, which is facilitated by a bi-directional gated recurrent unit. (2) a temporal multi-head self attention layer that captures dependencies across time steps in the input sequence allowing the model to attend to multiple subspaces of the input. (3) long term dependency decoder that decodes the complex feature representation learnt by the model using a temporal convolutional network. (4) an output network that facilitates multi-horizon quantile outputs. (5) Several gates that allows for skipping over superfluous components of the architecture. We show that the model achieves performance comparable to state-of-the-art forecasting performance by comparing the performance against several other approaches on two real-world datasets. We perform an ablation study showing the performance gains from each component of the model. Lastly we demonstrate the interpretability capabilities provided by the model and discuss how quantile outputs may be useful for decison makers. With regards to future research directions we would like to further optimize the architecture by exploring alternative feature encoding strategies such as discrete wavelet transforms. Another research direction would extending the model with the capability to process satellite images. Furthermore, exploring the capabilities provided by the attention module for explainable and interpretable forecasts is another potential research direction to pursue.

VIII. Acknowledgements

We would like to thank Kaixuan Chen for the supervison and guidance provided throughout the creation of this paper. We would also like to thank Thomas Fredholm, Thomas Kent and Mathias M. Andersen for the discussions that contributed to the development of this paper.

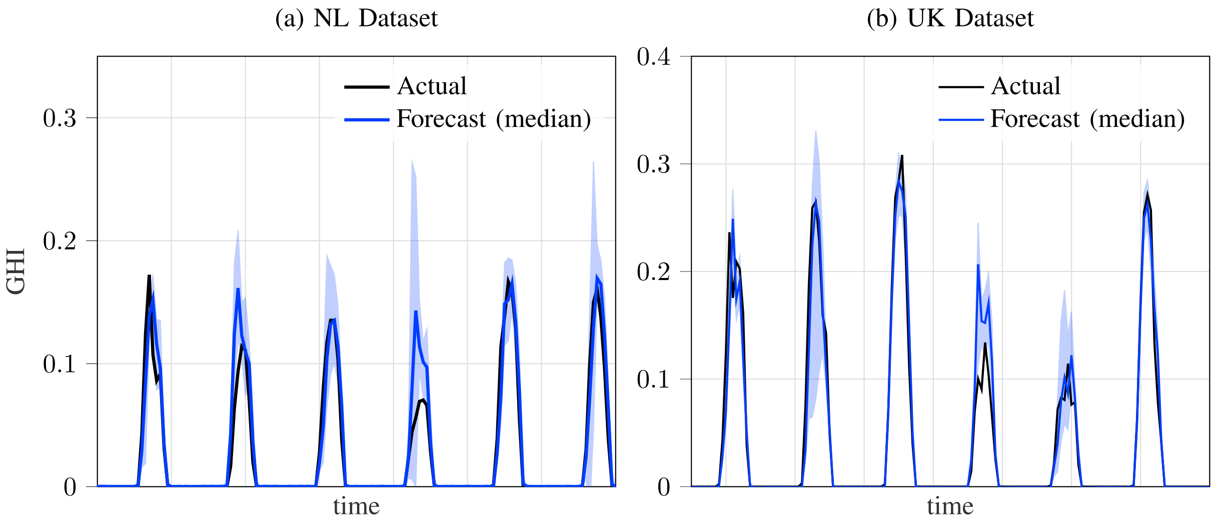

Figure 11: Excerpt of historical forecasts (Note: Normalized values. Light-blue shaded area indicate 98-% confidence interval)

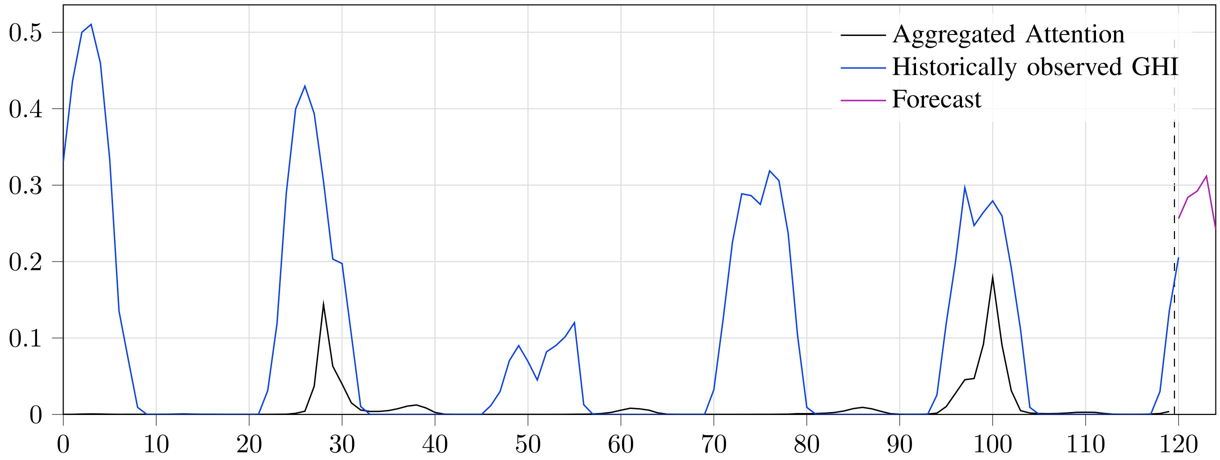

Figure 12: Highlight of important time steps in the input sequence.

References

[1] Huaizhi Wang et al. "A review of deep learning for renewable energy forecasting". In: *Energy Conversion and Management* 198 (2019), p. 111799. ISSN: 0196-8904. DOI: https://doi.org/10.1016/j.enconman.2019.111799. URL: https://www.sciencedirect.com/science/article/pii/S0196890419307812.

[2] Bella Espinar et al. "Photovoltaic Forecasting: A state of the art". In: *5th European PV-Hybrid and Mini-Grid Conference-Tarragona, Spain* (Apr. 2010).

[3] Moritz Stüber et al. "Forecast Quality of Physics-Based and Data-Driven PV Performance Models for a Small-Scale PV System". In: *Frontiers in Energy Research* 9 (2021). ISSN: 2296-598X. DOI: 10.3389/fenrg.2021.639346. URL: https://www.frontiersin.org/article/10.3389/fenrg.2021.639346.

[4] M. Boxwell. *Solar Electricity Handbook: A Simple, Practical Guide to Solar Energy : how to Design and Install Photovoltaic Solar Electric Systems*. Greenstream Publishing, 2012. ISBN: 9781907670183. URL: https://books.google.dk/books?id=dRrtwAEACAAJ.

[5] Supriyo Chakraborty et al. "Interpretability of deep learning models: A survey of results". In: *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. 2017, pp. 1–6. DOI: 10.1109/UIC-ATC.2017.8397411.

[6] Yu Zhang et al. "A Survey on Neural Network Interpretability". In: *IEEE Transactions on Emerging Topics in Computational Intelligence* 5.5 (2021), pp. 726–742. DOI: 10.1109/TETCI.2021.3100641.

[7] M. Schuster and K.K. Paliwal. "Bidirectional recurrent neural networks". In: *IEEE Transactions on Signal Processing* 45.11 (1997), pp. 2673–2681. DOI: 10.1109/78.650093.

[8] Bryan Lim et al. *Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting*. 2020. arXiv: 1912.09363 [stat.ML].

[9] Yitian Chen et al. *Probabilistic Forecasting with Temporal Convolutional Neural Network*. 2019. DOI: 10.48550/ARXIV.1906.04397. URL: https://arxiv.org/abs/1906.04397.

[10] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. DOI: 10.48550/ARXIV.1512.03385. URL: https://arxiv.org/abs/1512.03385.

[11] Yann N. Dauphin et al. *Language Modeling with Gated Convolutional Networks*. 2016. DOI: 10.48550/ARXIV.1612.08083. URL: https://arxiv.org/abs/1612.08083.

[12] Piotr Bojek and Heymi Bahar. *Solar PV – analysis*. Nov. 2021. URL: https://www.iea.org/reports/solar-pv.

[13] C.D. Winther. *Visual Guide to the Power Grid: Inside the Greatest Machine in the World*. Visual Power Grid Company. ISBN: 9788797195901. URL: https://books.google.dk/books?id=p1WmzQEACAAJ.

[14] Juan M. Morales et al. "Renewable Energy Sources—Modeling and Forecasting". In: *Integrating Renewables in Electricity Markets: Operational Problems*. Boston, MA: Springer US, 2014, pp. 15–56. ISBN: 978-1-4614-9411-9. DOI: 10.1007/978-1-4614-9411-9_2. URL: https://doi.org/10.1007/978-1-4614-9411-9_2.

[15] Pratima Kumari and Durga Toshniwal. "Deep learning models for solar irradiance forecasting: A comprehensive review". In: *Journal of Cleaner Production* 318 (2021), p. 128566. ISSN: 0959-6526. DOI: https://doi.org/10.1016/j.jclepro.2021.128566. URL: https://www.sciencedirect.com/science/article/pii/S0959652621027736.

[16] Rob Hyndman and G. Athanasopoulos. *Forecasting: Principles and Practice*. English. 3rd. Australia: OTexts, 2021.

[17] George.E.P. Box and Gwilym M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, 1976.

[18] Garazi Etxegarai et al. "An analysis of different deep learning neural networks for intra-hour solar irradiation forecasting to compute solar photovoltaic generators' energy production". In: *Energy for Sustainable Development* 68 (2022), pp. 1–17. ISSN: 0973-0826. DOI: https://doi.org/10.1016/j.esd.2022.02.002. URL: https://www.sciencedirect.com/science/article/pii/S0973082622000230.

[19] Chibuzor N. Obiora et al. "Forecasting Hourly Solar Radiation Using Artificial Intelligence Techniques". In: *IEEE Canadian Journal of Electrical and Computer Engineering* 44.4 (2021), pp. 497–508. DOI: 10.1109/ICJECE.2021.3093369.

[20] Chibuzor N Obiora, Ahmed Ali, and Ali N Hasan. "Implementing Extreme Gradient Boosting (XGBoost) Algorithm in Predicting Solar Irradiance". In: *2021 IEEE PES/IAS PowerAfrica*. 2021, pp. 1–5. DOI: 10.1109/PowerAfrica52236.2021.9543159.

[21] Kyairul Azmi Baharin et al. "Hourly irradiance forecasting in Malaysia using support vector machine". In: *2014 IEEE Conference on Energy Conversion (CENCON)*. 2014, pp. 185–190. DOI: 10.1109/CENCON.2014.6967499.

[22] Gabriel de Freitas Viscondi and Solange N. Alves-Souza. "A Systematic Literature Review on big data for solar photovoltaic electricity generation forecasting". In: *Sustainable Energy Technologies and Assessments* 31 (2019), pp. 54–63. ISSN: 2213-1388. DOI: https://doi.org/10.1016/j.seta.2018.11.008. URL: https://www.sciencedirect.com/science/article/pii/S2213138818301036.

[23] Leo Breiman. "Heuristics of instability and stabilization in model selection". In: *The Annals of Statistics* 24.6 (1996), pp. 2350–2383. DOI: 10.1214/aos/1032181158. URL: https://doi.org/10.1214/aos/1032181158.

[24] Dávid Markovics and Martin János Mayer. "Comparison of machine learning methods for photovoltaic power forecasting based on numerical weather prediction". In: *Renewable and Sustainable Energy Reviews* 161 (2022), p. 112364. ISSN: 1364-0321. DOI: https://doi.org/10.1016/j.rser.2022.112364. URL: https://www.sciencedirect.com/science/article/pii/S136403212200274X.

[25] Kenji Kawaguchi, Leslie Pack Kaelbling, and Yoshua Bengio. *Generalization in Deep Learning*. 2017. DOI: 10.48550/ARXIV.1710.05468. URL: https://arxiv.org/abs/1710.05468.

[26] Qiuxia Lai et al. "Understanding More About Human and Machine Attention in Deep Neural Networks". In: *IEEE Transactions on Multimedia* 23 (2021), pp. 2086–2099. DOI: 10.1109/TMM.2020.3007321.

[27] Gianni Brauwers and Flavius Frasincar. "A General Survey on Attention Mechanisms in Deep Learning". In: *IEEE Transactions on Knowledge and Data Engineering* (2021), pp. 1–1. DOI: 10.1109/TKDE.2021.3126456.

[28] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning Representations by Back-propagating Errors". In: *Nature* 323.6088 (1986), pp. 533–536. DOI: 10.1038/323533a0. URL: http://www.nature.com/articles/323533a0.

[29] Sepp Hochreiter and Jürgen Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (Nov. 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[30] Chibuzor N Obiora, Ahmed Ali, and Ali N Hasan. "Forecasting Hourly Solar Irradiance Using Long Short-Term Memory (LSTM) Network". In: *2020 11th International Renewable Energy Congress (IREC)*. 2020, pp. 1–6. DOI: 10.1109/IREC48820.2020.9310449.

[31] Fahad Radhi Alharbi and Denes Csala. "Short-Term Solar Irradiance Forecasting Model Based on Bidirectional

Long Short-Term Memory Deep Learning". In: *2021 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*. 2021, pp. 1–6. DOI: 10.1109/ICECCE52056.2021.9514233.

[32] Ashish Vaswani et al. *Attention Is All You Need*. 2017. DOI: 10.48550/ARXIV.1706.03762. URL: https://arxiv.org/abs/1706.03762.

[33] Abdelkader Dairi, Fouzi Harrou, and Ying Sun. "A deep attention-driven model to forecast solar irradiance". In: *2021 IEEE 19th International Conference on Industrial Informatics (INDIN)*. 2021, pp. 1–6. DOI: 10.1109/INDIN45523.2021.9557405.

[34] Aaron van den Oord et al. *WaveNet: A Generative Model for Raw Audio*. 2016. DOI: 10.48550/ARXIV.1609.03499. URL: https://arxiv.org/abs/1609.03499.

[35] Anjali Gautam and Vrijendra Singh. "Parametric Versus Non-Parametric Time Series Forecasting Methods: A Review". In: *Journal of Engineering Science and Technology Review* 13 (Jan. 2020), pp. 165–171. DOI: 10.25103/jestr.133.18.

[36] David Salinas, Valentin Flunkert, and Jan Gasthaus. *DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks*. 2017. DOI: 10.48550/ARXIV.1704.04110. URL: https://arxiv.org/abs/1704.04110.

[37] Ruofeng Wen et al. *A Multi-Horizon Quantile Recurrent Forecaster*. 2018. arXiv: 1711.11053 [stat.ML].

[38] Roger Koenker and Gilbert Bassett. "Regression Quantiles". In: *Econometrica* 46.1 (1978), pp. 33–50. ISSN: 00129682, 14680262. URL: http://www.jstor.org/stable/1913643 (visited on 05/16/2022).

[39] Kyunghyun Cho et al. *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches*. 2014. DOI: 10.48550/ARXIV.1409.1259. URL: https://arxiv.org/abs/1409.1259.

[40] Junyoung Chung et al. *Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling*. 2014. DOI: 10.48550/ARXIV.1412.3555. URL: https://arxiv.org/abs/1412.3555.

[41] Liang Li et al. "Temporal Attention Based TCN-BIGRU Model for Energy Time Series Forecasting". In: *2021 IEEE International Conference on Computer Science, Artificial Intelligence and Electronic Engineering (CSAIEE)*. 2021, pp. 187–193. DOI: 10.1109/CSAIEE54046.2021.9543210.

[42] Kaiming He and Jian Sun. *Convolutional Neural Networks at Constrained Time Cost*. 2014. DOI: 10.48550/ARXIV.1412.1710. URL: https://arxiv.org/abs/1412.1710.

[43] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. *Highway Networks*. 2015. DOI: 10.48550/ARXIV.1505.00387. URL: https://arxiv.org/abs/1505.00387.

[44] Yann N. Dauphin and David Grangier. *Predicting distributions with Linearizing Belief Networks*. 2015. DOI: 10.48550/ARXIV.1511.05622. URL: https://arxiv.org/abs/1511.05622.

[45] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. *Layer Normalization*. 2016. DOI: 10.48550/ARXIV.1607.06450. URL: https://arxiv.org/abs/1607.06450.

[46] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)*. 2015. DOI: 10.48550/ARXIV.1511.07289. URL: https://arxiv.org/abs/1511.07289.

[47] Zhouhan Lin et al. *A Structured Self-attentive Sentence Embedding*. 2017. DOI: 10.48550/ARXIV.1703.03130. URL: https://arxiv.org/abs/1703.03130.

[48] Romain Paulus, Caiming Xiong, and Richard Socher. *A Deep Reinforced Model for Abstractive Summarization*. 2017. DOI: 10.48550/ARXIV.1705.04304. URL: https://arxiv.org/abs/1705.04304.

[49] Chenyou Fan et al. "Multi-Horizon Time Series Forecasting with Temporal Attention Learning". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 2527–2535. ISBN: 9781450362016. DOI: 10.1145/3292500.3330662. URL: https://doi.org/10.1145/3292500.3330662.

[50] Colin Lea et al. "Temporal Convolutional Networks for Action Segmentation and Detection". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1003–1012. DOI: 10.1109/CVPR.2017.113.

[51] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling*. 2018. DOI: 10.48550/ARXIV.1803.01271. URL: https://arxiv.org/abs/1803.01271.

[52] Yong Shi, Wei Dai, and Wen Long. "A New Deep Learning-Based Zero-Inflated Duration Model for Financial Data Irregularly Spaced in Time". In: *Frontiers in Physics* 9 (2021). ISSN: 2296-424X. DOI: 10.3389/fphy.2021.651528. URL: https://www.frontiersin.org/article/10.3389/fphy.2021.651528.

[53] Rob Hyndman. "Another Look at Forecast Accuracy Metrics for Intermittent Demand". In: *Foresight: The International Journal of Applied Forecasting* 4 (Jan. 2006), pp. 43–46.

[54] *Global solar irradiance data*. 2022. URL: https://solcast.com/.

[55] Sjoukje Philip et al. "Regional differentiation in climate change induced drought trends in the Netherlands". In: *Environmental Research Letters* 15 (Sept. 2020). DOI: 10.1088/1748-9326/ab97ca.

[56] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. DOI: 10.48550/ARXIV.1412.6980. URL: https://arxiv.org/abs/1412.6980.

[57] Ilya Loshchilov and Frank Hutter. *SGDR: Stochastic Gradient Descent with Warm Restarts*. 2016. DOI: 10.

48550/ARXIV.1608.03983. URL: https://arxiv.org/abs/1608.03983.

[58] S. Gopal Krishna Patro and Kishore Kumar Sahu. *Normalization: A Preprocessing Stage*. 2015. DOI: 10.48550/ARXIV.1503.06462. URL: https://arxiv.org/abs/1503.06462.

[59] The pandas development team. *pandas-dev/pandas: Pandas*. Version latest. Feb. 2020. DOI: 10.5281/zenodo.3509134. URL: https://doi.org/10.5281/zenodo.3509134.

[60] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach et al. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[61] Julien Herzen et al. *Darts: User-Friendly Modern Machine Learning for Time Series*. 2021. arXiv: 2110.03224 [cs.LG].