big data and cognitive computing

MDPI

Article

An Efficient Green AI Approach to Time Series Forecasting Based on Deep Learning

Luis Balderas [1,2,3,4,*], **Miguel Lastra** [2,3,4,5] and **José M. Benítez** [1,2,3,4]

[1] Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain; j.m.benitez@decsai.ugr.es
[2] Distributed Computational Intelligence and Time Series Lab, University of Granada, 18071 Granada, Spain; mlastral@ugr.es
[3] Sport and Health University Research Institute, University of Granada, 18071 Granada, Spain
[4] Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain
[5] Department of Software Engineering, University of Granada, 18071 Granada, Spain
* Correspondence: luisbalru@ugr.es

Abstract: Time series forecasting is undoubtedly a key area in machine learning due to the numerous fields where it is crucial to estimate future data points of sequences based on a set of previously observed values. Deep learning has been successfully applied to this area. On the other hand, growing concerns about the steady increase in the amount of resources required by deep learning-based tools have made Green AI gain traction as a move towards making machine learning more sustainable. In this paper, we present a deep learning-based time series forecasting methodology called GreeNNTSF, which aims to reduce the size of the resulting model, thereby diminishing the associated computational and energetic costs without giving up adequate forecasting performance. The methodology, based on the ODF2NNA algorithm, produces models that outperform state-of-the-art techniques not only in terms of prediction accuracy but also in terms of computational costs and memory footprint. To prove this claim, after presenting the main state-of-the-art methods that utilize deep learning for time series forecasting and introducing our methodology we test GreeNNTSF on a selection of real-world forecasting problems that are commonly used as benchmarks, such as SARS-CoV-2 and PhysioNet (medicine), Brazilian Weather (climate), WTI and Electricity (economics), and Traffic (smart cities). The results of each experiment conducted objectively demonstrate, rigorously following the experimentation presented in the original papers that addressed these problems, that our method is more competitive than other state-of-the-art approaches, producing more accurate and efficient models.

Keywords: Green AI; dense feed-forward neural network simplification; time series forecasting

check for updates

Citation: Balderas, L.; Lastra, M.; Benítez, J.M. An Efficient Green AI Approach to Time Series Forecasting Based on Deep Learning. *Big Data Cogn. Comput.* **2024**, *8*, 120. https://doi.org/10.3390/bdcc8090120

Academic Editor: Domenico Ursino

Received: 31 July 2024
Revised: 23 August 2024
Accepted: 30 August 2024
Published: 11 September 2024

1. Introduction

Time series forecasting is a discipline that permeates almost every facet of society, playing a significant role in both industry and academia. It has countless applications in various fields where it is a key player, such as in economics and financial markets [1,2], climate prediction [3,4], energy consumption [5,6], and medical applications [7,8]. In all these areas, building sufficiently accurate artificial intelligence models is crucial for making future decisions [9].

In the past decade, the advent of deep learning, along with advances in hardware, has made it possible to develop increasingly sophisticated solutions that address both one-step-ahead and multi-horizon time series forecasting [10,11]. These solutions often rely on hybrid models that combine traditional statistical models with neural networks [12].

However, the current trend of enhancing the capabilities of these deep learning models focuses on increasing their size, whether in layers, neurons, or complexity, resulting in a rise in the number of parameters and floating-point operations. This leads to models that are very costly to train and maintain in terms of computation and memory footprint. Consequently, the use of these tools requires high levels of energy consumption, producing a notable adverse impact on the environment.

Contrary to this trend, in recent years, a paradigm called Green AI [13] has gained traction, aiming to continue the evolution of technology while minimizing environmental impact. In this article, we present a methodology to build deep learning models for time series forecasting following the Green AI paradigm, called GreeNNTSF (Figures 1 and 2). Specifically, we propose a semi-automatic method for building efficient neural networks. The ODF2NNA algorithm [14] is one of the fundamental pillars of our methodology, as it automates the network simplification process. We demonstrate that, compared to state-of-the-art techniques recognized as references for these problems due to their recent publications, it is possible to build solutions that are not only more efficient but also more effective. To achieve this, we use prediction metrics, closely following the experimentation outlined in the reference articles, as well as efficiency metrics, such as the reduction in the number of model parameters. The objective of this article is to apply the methodology we present to time series forecasting in order to construct models that effectively solve widely prevalent real-world problems related to economics, climate, smart cities, and medicine with the minimum number of parameters possible. Our main contributions can be summarized as follows:

- We developed an end-to-end methodology to build Green AI-based deep learning models for time series forecasting.
- We applied this methodology to six real-world problems in economics, medicine, smart cities, and climate, generating models that outperform the state-of-the-art methods in terms of precision and efficiency.

The structure of the article is as follows. In Section 2, we present the Green AI paradigm and the state-of-the-art methods that use deep learning for time series forecasting. In Section 3, we describe our methodology. In Section 4, we detail our empirical analysis and discussion. Finally, Section 5 contains the conclusions of the work.

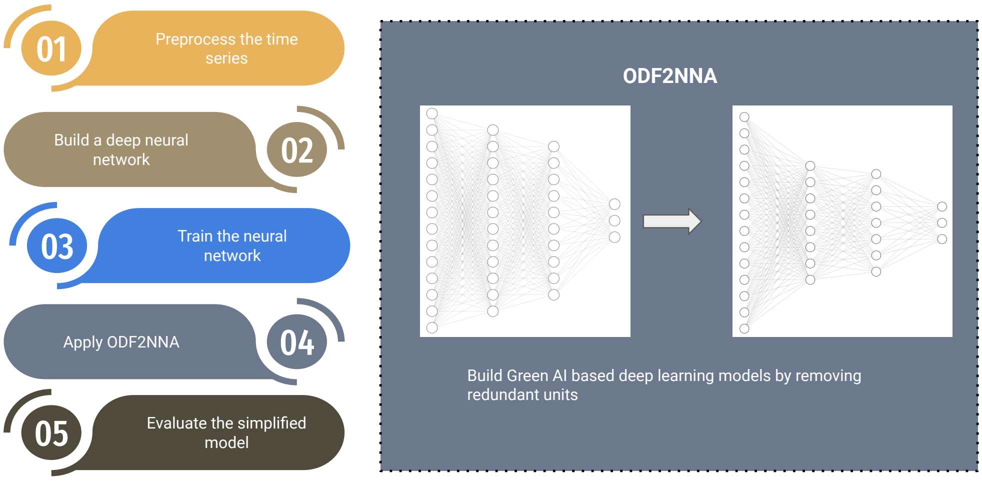
01
Preprocess the time series
02
Build a deep neural network
03
Train the neural network
04
Apply ODF2NNA
05
Evaluate the simplified model
ODF2NNA
Build Green AI based deep learning models by removing redundant units

Figure 1. GreeNNTSF is an end-to-end methodology to build deep learning models based on the Green AI paradigm for time series forecasting.

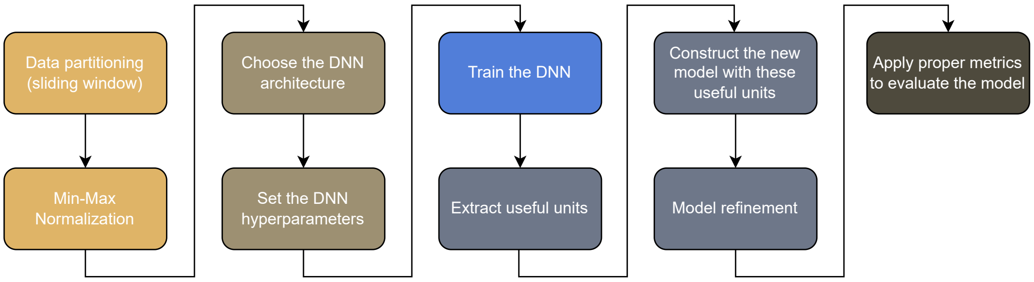
Data partitioning (sliding window)
Min-Max Normalization
Choose the DNN architecture
Set the DNN hyperparameters
Train the DNN
Extract useful units
Construct the new model with these useful units
Model refinement
Apply proper metrics to evaluate the model

Figure 2. GreeNNTSF explained in detail. Time series preprocessing includes the data partitioning and the normalization stages. After that, a DNN is built by choosing an architecture as the starting point and setting its hyperparameters; next, the resulting DNN is trained. After that, ODF2NNA is applied by extracting the useful neurons, constructing a new simplified model including only these, and applying a short model refinement. Finally, the simplified model is evaluated using time series forecasting and Green AI metrics.

2. Related Work

This section is dedicated, first, to introducing the fundamental concepts of Green AI as well as some metrics that justify the need for these techniques. Subsequently, we present some of the latest state-of-the-art methods in time series forecasting. These methods are applied to various sectors or industries to address real-world problems.

2.1. Green AI

As stated in [13], the term Green AI refers to AI research aimed at achieving scientific advancements while considering computational costs and aiming to reduce the resources required for development. One of the fundamental principles underpinning this paradigm is the balance between efficiency and effectiveness. To this end, when scientific advancements occur, such as new models for solving problems, not only are effectiveness metrics considered, such as accuracy, F1-Score, or precision in classification, among others, but efficiency measures also take a prominent role. Efficiency metrics focus on measuring the amount of work, translated to computational resources, required for tasks such as training AI models or using them for inference. There are various efficiency measures, including carbon emissions, electricity usage, the time taken to generate the model, and the number of parameters comprising it (which also serves as a measure of complexity). Henceforth, we will use the number of parameters as a reference efficiency metric, given that it is a metric that is easily measured (as opposed to others that exceed software capabilities, such as emissions or energy consumption) and is widely accepted and used in the literature.

The development of deep learning, combined with advances in hardware, is revolutionizing industry, academia, and the daily lives of citizens. Since the emergence of the Transformer architecture [15], generative AI models have been created that solve use cases in radically new ways, achieving unprecedented levels of quality and accuracy. However, training these models (and their subsequent use) incurs very high CO_2 emission costs. As shown in [16], training GPT-3 ([17]), with 175 billion parameters, resulted in the emission of 502 tons of CO_2, which is eight times the emissions of a vehicle's entire lifespan or nearly 100 times more than what an average human emits in a year. Additionally, in terms of energy consumption, Ref. [18] estimated that training GPT-3 required an amount of energy equivalent to the consumption of 121 homes in the U.S. (1287 MWh). Nonetheless, these costs are not restricted to training, taking into account that, in inference time, each ChatGPT (with GPT-4 as the backbone model) [19] query consumes 260.42 MWh per day [16]. Therefore, we must continue developing AI models that drive technological advancement while minimizing environmental impact. According to [16], the first step in building Green AI models is the optimization of algorithms by designing methods that reduce the amount of resources needed for training or usage. Neural network pruning is one of the most widely used techniques in the literature due to its effectiveness in reducing network complexity and, consequently, associated computational costs.

2.2. Some Deep Learning Methods for Time Series Forecasting

Time series forecasting is a widely practiced discipline that applies to numerous academic and industrial problems and has been approached from multiple perspectives. These approaches range from classical methods, such as autoregressive (AR) models, autoregressive moving average (ARMA) models, and autoregressive integrated moving average (ARIMA) models, to machine learning models like random forests or support vector machines (SVMs). With the rise of deep learning, neural networks have become the reference solution, whether they are classical dense feed-forward networks or other types, such as recurrent neural networks (RNNs) in their various forms, like long short-term memory (LSTM) or gated recurrent units (GRUs), and convolutional neural networks (CNNs). In recent years, the Transformer architecture has been adopted due to the similarity between time series forecasting and language generation models. Intuitively, in both cases, there is a specific order in which the elements of a sentence (or a time series) are expressed to ensure coherence, such that the next word or value in a series largely depends on the preceding ones. This makes it natural to use similar technologies in solving these two seemingly different problems.

This work does not aim to exhaustively cover all approaches to time series forecasting. For an extensive review of recent developments in this field, see [12]. However, we delve into certain recently published methods that provide different solutions to various problems and that have been widely adopted as benchmarks in time series forecasting. These state-of-the-art problems and methods are used in this work to compare and assess the effectiveness of our approach.

In [20], a new neural network architecture for time series forecasting called Recurrent Graph Evolution Neural Network (REGENN) is presented, combining graph evolution and deep recurrent learning. Specifically, REGENN consists, in parallel, of a linear component with a feed-forward layer and a nonlinear component with an autoencoder. In the encoder part, a non-conventional Transformer encoder is introduced, followed by a graph soft evolution layer. The decoder uses two sequence-to-sequence layers, specifically LSTM. REGENN utilizes three multivariate datasets to measure its effectiveness. The first, related to medicine, comes from Johns Hopkins University and focuses on SARS-CoV-2 [21]. The second, related to climate, is the Brazilian weather dataset. Finally, the third, also from the medical field, corresponds to the 2012 PhysioNet Computing in Cardiology Challenge [22]. As can be intuitively observed in Figures 5, 7, and 9 of Ref. [20] (exact values are not provided), REGENN achieves better results than other state-of-the-art techniques.

In [23], the problem of multistep time series forecasting for non-stationary signals with sudden changes is addressed. To this end, the authors propose incorporating temporal and shape criteria in the training process, defining similarity and difference measures for both concepts. Using dynamic time warping (DTW) and the temporal distortion index (TDI), the article presents a new objective function called DILATE for probabilistic forecasting and a prediction framework called STRIPE++. STRIPE++ is based on an autoencoder, where the encoder summarizes the input into a latent vector. This vector is then transformed by the decoder into a trajectory, which facilitates the capture of variations in the upcoming values of the time series. STRIPE++ has been used on synthetic and electricity datasets mentioned in the paper, which we could not access, and on the publicly available Traffic dataset, obtaining better results than other techniques, such as Diverse DPP [24], Deep AR [25], Nbeats [26], and Variety Loss [27].

Another paper that designs a deep learning-based approach for predicting electricity demand (using the Electricity ENTSO-E dataset) is [28]. By combining exponential smoothing (ES) and recurrent neural networks, they present the ES-dRNN method. The ES component dynamically extracts the components of each individual time series. As a result, each time series is deseasonalized, normalized, and squashed. After that, a multilayer RNN, along with a new type of neural unit called a dilated recurrent cell, extracts short- and long-term dependencies within the time series. Additionally, the model generates probability intervals to express the uncertainty of the predictions. ES-dRNN has been compared to other state-of-the-art methods, such as ARIMA, K-Nnw, Prophet, LSTM, and SVM, among others, obtaining more accurate results.

Finally, in [29], the problem of crude oil price forecasting is addressed. Using the West Texas Intermediate (WTI) daily closing oil prices from 2 August 2010 to 31 December 2019, two hybrid methods based on RNN using variational mode decomposition (VMD), sample entropy (SE), and gated recurrent units (GRUs) are presented. They point out that, for predicting oil prices, GRU models perform exceptionally well, outperforming LSTM and DNN in terms of accuracy and execution time. Moreover, the combination of VMD, SE, and GRU achieves competitive and robust results.

3. Our Proposal

In this article, we propose GreeNNTSF (Figures 1 and 2), a Green AI-based methodology for time series forecasting that constructs deep learning models that are more environmentally efficient without losing predictive effectiveness. Contrary to the trend of building increasingly larger networks to solve problems, the core idea of our approach lies in constructing a use case that includes the semi-automatic construction of a feed-forward neural network. This model is then simplified, as our methodology allows for the automatic discovery of a subnetwork within the original neural network, resulting in a simplified model with fewer parameters without compromising the quality of the predictions.

Concretely, our methodology is an end-to-end process that starts with the problem, expressed through a dataset, that we aim to solve. First, we preprocess the instances and construct the final dataset by applying the sliding window technique, which involves transforming a time series from its original form, with values one after another, into a set of instances for training models. Specifically, by setting a time window, the window slides over the series, taking subsets of the same size from left to right. Each subset becomes an instance for the future model, consisting of the input used for training and the expected values to forecast.

Next, we propose a dense feed-forward neural network. Given that choosing the architecture, including layers and neurons per layer, is a complex task, our methodology favors simple network topologies with a sufficient number of layers and neurons. For simplicity of use, no hyperparameter tuning related to the neural network (for example, learning rate) is required in our methodology.

After training the network, we apply the pruning algorithm ODF2NNA [14], a simplification algorithm for dense feed-forward neural networks based on pruning. It is very easy to use, as it only has one tolerance parameter, ϵ, which indicates the pruning intensity. In this way, different versions of a more or less simplified model can be obtained. The algorithm works by measuring the redundancy of each neuron with respect to each instance in a dataset. If the redundancy level exceeds the tolerance, the neuron being evaluated is considered irrelevant to the prediction for that instance and can be removed. In contrast, if it has a low redundancy level, the neuron's contribution is deemed useful and is retained in the simplified model. Given that every neuron has to be evaluated to decide whether it is included in the simplified model or not, ODF2NNA has linear complexity with respect to the number of neurons in the original network.

ODF2NNA is compatible with both classification and regression problems, adapting to canonical metrics such as classification accuracy, F1-Score, or MSE, allowing for quantitative measurement of the predictive ability of the simplified model. For the sake of clarity, the following key points show the ODF2NNA's main ideas:

- Since it has a trained original model as input, ODF2NNA's objective is to build a simplified version of the original one.
- The simplification phase starts by extracting the useful units per layer from the original model by applying the tolerance parameter ϵ.
- The greater ϵ is, the more intense is the pruning.
- Once the extraction finishes, the simplified model is built including only the useful neurons found during the process.

- Finally, the simplified model is refined with a light training process.

The resulting simplified model, adhering to Green AI standards, is ready to be used as a forecasting engine in production. The methodology is presented in Algorithm 1.

Algorithm 1 Description of the GreeNNTSF Methodology

```
function GREENNTSF($ts$, $\epsilon$)
    1. Preprocess the original time series, $ts$. Use sliding window technique to build the definite version of our time series, $tsf$.
    2. Construct a dense feed-forward neural network for addressing the $tsf$ time series forecasting task.
    3. Train the neural network
    4. Apply ODF2NNA: build a new model by pruning and refining the original model with $\epsilon$ network reduction level.
    5. Evaluate the simplified model using established benchmarks.
end function
```

In summary, GreeNNTSF is domain-agnostic and robust technique, as it can tackle real-world time series forecasting problems across various domains, impacting both industry and academia. It generates optimized models following the principles of Green AI. Additionally, it does not impose any restrictions on the problem to be solved, nor does it require specific data preprocessing, making it completely flexible and adaptable to the needs of each use case.

To measure the effectiveness of our methodology, we have selected a set of real-world problems from various sectors recognized in the literature as benchmarks for time series forecasting. The empirical analysis conducted is detailed in the following section.

4. Empirical Analysis

In this article, a methodology is proposed for building time series forecasting models following the Green AI paradigm. We have thoroughly evaluated our scheme considering real-world problems in fields such as economics, climate, smart cities, and medicine. The section is organized as follows: first, the different datasets that embody the problems to be solved are presented. Next, the metrics used to evaluate the suitability of the solution are introduced. Both the datasets and metrics have been meticulously chosen following those used as benchmarks in the latest state-of-the-art publications in these fields. Subsequently, more details are provided about the procedure for constructing the neural networks and their subsequent simplification. Finally, the results obtained are discussed.

4.1. Real-World Problems

For this work, datasets representing real-world problems have been chosen. Due to their diversity and the different approaches published in the state of the art for their resolution, these datasets constitute an appropriate benchmark for evaluating novel models. As mentioned, the problems described here are of interest to both industry and academia, being associated with economics, smart cities, medicine, or climate. The datasets are as follows:

- SARS-CoV-2: Dataset provided by John Hopkins University [21]. It consists of three variables measured over 120 days in 188 different countries, representing the first four months of the pandemic. The features are the number of recovered patients, the number of infected patients, and the number of deaths.
- Brazilian Weather: Dataset generated by collecting data from 253 sensors over 1095 days [20]. It consists of four variables: minimum temperature, maximum temperature, solar radiation, and rainfall.
- PhysioNet: Dataset created for the 2012 PhysioNet Computing in Cardiology Challenge [22]. It consists of nine features collected over 48 h from 11,988 ICU patients. The features are non-invasive diastolic arterial blood pressure, non-invasive systolic arterial blood pressure, invasive diastolic arterial blood pressure, non-invasive mean

arterial blood pressure, invasive mean arterial blood pressure, urine output, heart rate, and weight.

- WTI: One of the most widely used benchmarks in the global oil market, available from the U.S. Energy Information Administration [30]. For this experiment, 2446 WTI closing prices from 2 August 2010 to 31 December 2019 are included.
- Traffic: Dataset composed of the road occupancy rate from the California Department of Transportation, measured every hour over 48 months from 2015–2016 [23].
- Electricity: Includes the electricity demand of 35 European countries between 2016 and 2018, available at the ENTSO-E webpage [31].

To ensure the experiments are comparable, the same training and test partitions indicated in each of the publications used for comparison have been employed. For full details on these partitions, please refer to the original publications.

4.2. Metrics

The metrics, like the datasets, have been rigorously selected according to those used in the articles presenting the techniques with which our time series forecasting method is compared. Most of them are classic in this discipline, such as mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE) or R^2. Others are somewhat less conventional, such as mean squared logarithmic error (MSLE) and median absolute percentage error (MdAPE) [32,33]. Finally, some metrics are defined ad hoc in the reviewed articles, such as DILATE: DIstortion Loss including shApe (based on dynamic time warping or DTW [34]) and TimE (based on the time distortion index or TDI for temporal misalignment estimation [35]). DILATE is defined as follows. If $\hat{y}$ is the prediction and y^* the ground truth:

$$L_{\text{DILATE}}(\hat{y}, y^*) = \alpha L_{\text{Shape}}(\hat{y}, y^*) + (1-\alpha) L_{\text{Time}}(\hat{y}, y^*),$$

with

$$L_{\text{Shape}} = \text{DTW}(\hat{y}, y^*),$$
$$L_{\text{Time}} = \text{TDI}(\hat{y}, y^*).$$

More details can be found in [23]. Table 1 summarizes the correspondence between datasets and error metrics used to compare different time series forecasting methods.

Table 1. Metrics used for each learning task or dataset. This helps in understanding how each dataset's forecasting performance is assessed using various error metrics, ensuring a comprehensive evaluation of the proposed models against existing state-of-the-art techniques.

Learning Task/Dataset	**Metrics**
SARS-CoV-2	MAE, RMSE, MSLE
PhysioNet	MAE, RMSE, MSLE
Brazilian Weather	MAE, RMSE, MSLE
Traffic	MSE, DILATE
Electricity	MAPE, RMSE, MdAPE
WTI	RMSE, MAE, MAPE, R^2

To apply the mentioned metrics, implementations available in specialized time series software packages, such as sktime (v0.32.4) [36,37], are used when available. For DILATE, the implementation provided by the authors in their article is used. By using standard implementations and rigorously applying the criteria presented in the original works, we ensure that the comparison is fair and useful for drawing unbiased conclusions.

4.3. Dense Feed-Forward Neural Network Definition and Simplification

One of the major strengths of our methodology for time series forecasting is the simplicity of the networks we use. Unlike state-of-the-art reference methods, our neural networks are dense feed-forward networks with a relatively simple structure. Our approach is designed to be versatile, accommodating problems of very different natures. Consequently, the architecture and number of parameters of each network vary depending on the specific problem. However, the overall procedure remains consistent: a dense feed-forward neural network is designed to solve a supervised learning task. This network is then simplified using ODF2NNA [14], resulting in a smaller network without losing accuracy—sometimes even improving it. After applying the algorithm, the simplified network is compared with state-of-the-art techniques to evaluate whether our solution is competitive.

Below are the different neural network architectures and configurations for each of the datasets described in Section 4.1. As mentioned before, simple yet effective neural network architectures are chosen as part of our process in order to be simplified afterwards. Next, we present the original architecture proposed (before simplification) for each of the addressed problems:

- SARS-CoV-2: Neural network with 6 layers: 300-200-100-50-20-3.
- Brazilian Weather: Neural network with 7 layers: 300-200-100-50-25-12-4.
- PhysioNet: Neural network with 7 layers: 500-300-200-100-50-20-9.
- WTI: Neural network with 6 layers: 300-200-100-50-20-1.
- Traffic: Neural network with 6 layers: 300-200-100-50-20-1.
- Electricity: Neural network with 6 layers: 500-250-200-100-50-24.

With respect to the hyperparameters, ReLU is chosen as the activation function for all layers except for the output layer. Additionally, Adam is set as the optimizer and the mean square error as the loss function. The learning rate is set to 0.001, β_1 to 0.9, and β_2 to 0.99.

Finally, one of the main limitations of our method is the selection of the original neural network from which the simplified model is constructed. If this model is poorly designed, the rest of the process will not yield satisfactory results. For instance, if the network is too small, it will be unable to learn properly, leading to underfitting issues. Conversely, if it is too large relative to the forecasting problem to be solved, the model will suffer from overfitting. However, after applying the simplification process, the overfitting issue might be mitigated. In any case, it is crucial to design the network well and to ensure that subsequent training generates satisfactory results before applying the simplification. Nonetheless, this is an intrinsic limitation of all methods involving neural networks, as no formal procedure has yet been defined to unequivocally determine the best network topology for a specific problem.

4.4. Results and Discussion

In this section, we present the results obtained for each of the experiments conducted. The objective is to assess the extent to which our approach based on Green AI is more suitable for time series forecasting problems. The results shown in the tables correspond to those generated by the models after the complete methodology has been applied. This means that the network is trained, simplified, and finally used to predict the test set for each of the learning tasks.

In all experiments, we have strictly followed the guidelines of the original articles for data preprocessing as well as partitioning of the data by applying the sliding window technique for executing the corresponding trainings and validations. All approaches agree on using min-max normalization to achieve better training results. In every case, the training set was used to adjust the normalization, which was then applied to the test set. Finally, the model outputs were transformed back to their original scale before calculating the performance metrics.

The model simplification stage, presented in Section 4.3, involves the application of the ODF2NNA method. This requires selecting the ϵ parameter, which represents the tolerance level in pruning. This value is closely related to the specific problem being addressed. The higher the tolerance value, the more intensive the pruning, resulting in more simplified models. For each problem, a grid search is conducted with a limited number of values (approximately 5 different values), and the model that achieves the best results is chosen as the simplified model.

The results of each experiment are presented and discussed below. Since we aggregate real-world problems addressed in various state-of-the-art articles, for the sake of clarity, we distinguish each experiment in a separate section. If multiple experiments are covered in the same reference article, they are addressed in the same section due to the similarity in the techniques used.

4.4.1. Experiments on the SARS-CoV-2, Brazilian Weather, and PhysioNet Datasets

In this section, we present the results for experiments on the SARS-CoV-2, Brazilian Weather, and PhysioNet datasets. For all of them, the most recent state-of-the-art technique is REGENN [20].

The results of the experiment on the SARS-CoV-2 dataset are shown in Table 2. Given that the dataset contains information from 188 different countries, in contrast to REGENN, which proposes a single model, our approach involves creating a model for each country to generate results that are much more tailored to the specific circumstances of each region. Therefore, the results shown in the table represent, in our case, the average of the metrics obtained by the ensemble model for each test set.

Table 2. Results for the SARS-CoV-2 experiment. MAE stands for mean absolute error, RMSE stands for root mean square error, and MSLE stands for mean squared logarithmic error. Although REGENN performs better in terms of the MSLE metric, our approach is much more accurate when considering metrics such as MAE and RMSE, with the latter being notably lower compared to other state-of-the-art methods. The best results are shown in bold.

Algorithm	MAE	RMSE	MSLE
REGENN [20]	165.41	915.92	**0.05**
GreeNNTSF	**134.13**	**174.25**	0.1

With respect to the data preparation, a window size of 7 days is used for the training set, with 7 days reserved for validation. Finally, the goal is set to predict the values for the next 14 days. This 7-7-14 configuration follows medical expert knowledge regarding the virus incubation period, as stated in the original article.

The original work includes a graphical comparison of the reference method against other time series forecasting models, such as linear, autoregressive, LSTM, GRU, decision tree, k-nearest neighbors, random forest, and exponential smoothing approaches, among others. Although the exact metric values cannot be precisely extracted, it is clear that REGENN outperforms the other techniques for the MAE, RMSE, and MSLE metrics. However, as summarized in Table 2, we obtain a much more compact and efficient ensemble, given the fact that it achieves better results in both MAE (19% reduction) and RMSE (81% reduction), showing a notable reduction compared to the other experiments and especially against REGENN, which is the reference. For MSLE, we also obtain competitive results, although REGENN achieves the best performance.

The results of the experiment on the Brazilian Weather dataset are shown in Table 3. In this case, the dataset contains information from 253 different sensors. Thus, a model for each sensor is built to obtain more accurate predictions. With respect to the data preparation, following the seasonality of weather data, an 84-day window size is used, reserving 28 days for validation and predicting the next 56 days. Similar to the last experiment, the results shown in the table represent, in our case, the average of the metrics obtained by the ensemble model for each test set. GreeNNTSF obtains the best results for all metrics, obtaining a 13.5% reduction for MAE, 40% reduction for RMSE, and 17% for MSLE.

Table 3. Results for the Brazilian Weather experiment. MAE stands for mean absolute error, RMSE stands for root mean square error, and MSLE stands for mean squared logarithmic error. In this case, our methodology obtains the best results for all metrics. The best results are shown in bold.

Algorithm	MAE	RMSE	MSLE
REGENN	1.92	4.86	0.28
GreeNNTSF	**1.68**	**2.91**	**0.23**

Finally, Table 4 presents the results of the PhysioNet dataset. To learn the behavior of the 11,988 ICU patients represented, an ensemble of 11,988 models is generated. The window size, following the guidelines of the original article, is set to 12 h, with 6 h reserved for validation, and the objective is to predict the next 6 h of the dataset. Unlike previous datasets, a much shorter time period is established to track the patient hour by hour, rather than making long-term predictions. It can be observed that the results of our methodology are more competitive than the reference algorithm and the other proposed approaches (19% reduction for MAE, 14% for RMSE, and 36% for MSLE).

Table 4. Results for the PhysioNet experiment. MAE stands for mean absolute error, RMSE stands for root mean square error, and MSLE stands for mean squared logarithmic error. Again, GreeNNTSF obtains the best results for all metrics. The best results are shown in bold.

Algorithm	MAE	RMSE	MSLE
REGENN	18.22	47.31	1.37
GreeNNTSF	**14.77**	**40.52**	**0.88**

4.4.2. Experiment on the WTI Dataset

Given that WTI crude oil is one of the primary price benchmarks in the global oil market, accurately predicting its value provides a significant competitive advantage, making it a highly relevant use case in economics. For setting up the experimentation from a data perspective, 2346 closing price values of WTI are taken, with the last 100 values reserved as the test set. We define the problem as a one-step-ahead forecasting task with a window size of 3. As mentioned in [29], the data are smooth, autocorrelated, and do not follow a normal distribution, containing a substantial amount of noise.

As shown in Table 5, Ref. [29] proposes various approaches that combine variational modal decomposition, sample entropy, and gated recurrent units. There are no details provided regarding the number of parameters these different models have or indications of their complexity, so we cannot compare the results in that aspect. Nonetheless, we can see that GreeNNTSF yields better results in terms of the RMSE, MAPE, and R^2 metrics, although the VMD-GRU configuration achieves a better MAE value. Finally, it can be verified that our method is effective, as there is another approach based on dense neural networks that yields significantly higher error rates (two to three times higher).

4.4.3. Experiment on the Traffic Dataset

In the experiment with the Traffic dataset, the ratio of road occupancy in California is measured hourly. Following the guidelines of the reference article [23], the time series, consisting of 17,544 values, is divided by using 60% for training, the next 20% for validation, and the remaining 20% for testing. The objective is to predict the next 24 values, that is, the occupancy rate for the next day measured hour by hour, given the previous 168 points, which represent the traffic from the previous week.

Table 5. Results for predicting the WTI price. RMSE stands for root mean square error, MAE stands for mean absolute error, MAPE stands for mean absolute percentage error, and R^2 is the correlation coefficient. As shown, GreeNNTSF achieves better results across all metrics except for MAE, where VMD-GRU has the best result. The best results are shown in bold.

Algorithm	RMSE	MAE	MAPE	R^2
LSTM	1.402	0.835	1.646	0.685
GRU	1.268	0.816	1.43	0.742
DNN	1.481	0.962	1.723	0.626
BPNN	1.666	1.213	2.131	0.554
VDM-LSTM ([29])	0.681	0.462	0.815	0.825
VMD-GRU	0.654	**0.429**	0.756	0.931
VMD-SE-LSTM (univariate)	0.817	0.556	0.982	0.892
VMD-SE-LSTM (multivariate)	0.796	0.532	0.942	0.898
VMD-SE-GRU (univariate)	0.669	0.451	0.793	0.928
VMD-SE-GRU (multivariate)	0.783	0.523	0.924	0.901
GreeNNTSF	**0.612**	0.498	**0.587**	**0.932**

In [23], the authors do not provide details on the number of parameters of STRIPE++, although they explain that it is an autoencoder and that it specifies the number of GRU units and dense layers. Therefore, once again, we cannot compare the number of parameters between our approach and their algorithm. Table 6 displays the prediction metric results. For both MSE, a classic forecasting metric, and DILATE, introduced in the reference, GreeNNTSF achieves better results compared to other state-of-the-art methods.

Table 6. Results for the Traffic dataset learning task. MSE stands for mean square error, and DILATE is introduced in Section 4.2. The lower the value, the better the model's performance. As shown, our approach yields better results than the other methods, including STRIPE++. The best results are shown in bold.

Algorithm	MSE	DILATE
STRIPE++ [23]	6.7	14.1
Diverse DPP [24]	6.9	14.7
Deep AR [25]	6.6	16.9
Nbeats [26]	17.1	17.8
Variety Loss [27]	7.9	15.9
GreeNNTSF	**4.78**	**12.46**

4.4.4. Experiment on the Electricity Dataset

The experiment on Electricity dataset serves as a clear example of an applied problem in economics. The reference algorithm ES-eRNN [28] proposes a complex model that combines exponential smoothing and recurrent neural networks, with approximately 229,000 parameters. Additionally, note that there are two proposals: ES-dRNN with a simple model and ES-dRNNe with an ensemble. The objective is to predict the next 24 values, with a window size of 192 values. The results are shown in Table 7. It can be observed that the reference model performs better than the other machine learning models proposed in the original work. However, our methodology achieves even better results across all three metrics (5% reduction for MAPE, 20% reduction for RMSE, and 54% for MdAPE with respect to ES-dRNNe) with a simplified model. Initially starting with 280,000 parameters, we reduced this number to 184,207, which represents a 20%

reduction with respect to ES-dRNNe. Therefore, we not only generate a model with better performance but also make it more efficient, aligning with the principles of Green AI.

Table 7. Results for predicting demand in the Electricity dataset. RMSE stands for root mean square error, MAE stands for mean absolute error, MAPE stands for mean absolute percentage error, and MdAPE stands for median absolute percentage error. As we can see, our methodology outperforms all other state-of-the-art methods in every metric. The best results are shown in bold.

Algorithm	MAPE	RMSE	MdAPE
Naive	5.08	704.34	4.84
ARIMA ([38])	3.3	475.09	3.01
ES ([38])	3.11	439.26	2.88
Prophet ([39])	4.53	619.39	4.32
K-Nnw ([38])	2.5	335.13	2.31
FNM ([38])	2.5	334.08	2.3
N-WE ([38])	2.49	332.49	2.28
GRNN ([40])	2.48	332.91	2.28
MLP ([40])	3.05	419.01	2.78
SVM ([41])	2.55	357.24	2.29
LSTM ([42])	2.76	381.76	2.57
ANFIS ([43])	3.65	507.08	3.17
MTGNN ([44])	2.99	405.18	2.74
ES-dRNN ([28])	2.36	326.06	2.15
ES-dRNNe ([28])	2.23	306.94	2.02
GreeNNTSF	**2.16**	**251.12**	**0.94**

5. Conclusions

Time series forecasting is a discipline with enormous applications in both industry and academia. The literature presents numerous approaches that address problems of very different natures. The advent of deep learning has led to increasingly powerful solutions. However, the models generated tend to be very large, which, despite being very accurate, entail significant computational, energy, and environmental costs. Green AI is a paradigm that aims to introduce models with high predictive capacity that are more resource-efficient. In this work, we propose GreeNNTSF, an end-to-end methodology to build Green AI time series forecasting models that address real-world problems in economics, medicine, smart cities, and climate. Through meticulous experimentation, including different datasets and forecasting metrics, we demonstrate that the models generated by our methodology are not only more efficient due to reducing the number of parameters in their architecture (e.g., 20% reduction with respect to the state-of-the-art in the Electricity experiment) but also more effective in prediction, outperforming other state-of-the-art approaches in almost all metrics.

As future work, on the one hand, we intend to leverage the power of our method to create hybrid models that combine time series data with statistical features of the series as input to the neural network. This approach could generate deep learning models under the Green AI paradigm, combining the strong predictive capabilities of deep learning models with expert knowledge from time series feature extraction methods. On the other hand, we would like to explore other types of neural networks, such as convolutional or Transformer models, for time series forecasting with a Green AI approach.

Author Contributions: Conceptualization: L.B., M.L. and J.M.B.; methodology: L.B., M.L. and J.M.B.; software: L.B.; validation: L.B.; investigation: M.L. and J.M.B.; writing—original draft: L.B.; writing—review and editing: M.L. and J.M.B.; visualization: L.B.; supervision: M.L. and J.M.B.; project administration: J.M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been partially supported by the Spanish Ministry of Economy, Industry, and Competitiveness (PID2023-151336OB-I00) with the co-financing of the European Union (FEDER). Proyecto PID2020-118224RB-I00 financiando por Ministerio de Ciencia, Innovación y Universidades, MICIN/AEI/10.13039/501100011033.

Data Availability Statement: Data available in a publicly accessible repository. Specifically (accessed on 30 July 2024):

- Brazilian Weather. URL/DOI: https://github.com/gabrielspadon/ReGENN/tree/master/data, https://doi.org/10.1109/TPAMI.2021.3076155.
- Sars-Cov-2. URL/DOI: https://github.com/gabrielspadon/ReGENN/tree/master/data, https://doi.org/10.1109/TPAMI.2021.3076155, https://doi.org/10.1016/s1473-3099(20)30120-1.
- PhysioNet. URL/DOI: https://github.com/gabrielspadon/ReGENN/tree/master/data, https://doi.org/10.1109/TPAMI.2021.3076155, http://www.cinc.org/archives/2012/pdf/0245.pdf.
- Traffic. URL/DOI: https://doi.org/10.1109/TPAMI.2022.3152862, https://data.ca.gov/dataset/traffic-volumes-aadt.
- WTI. URL/DOI: www.eia.doe.gov, https://doi.org/10.1016/j.eswa.2023.119617.
- Electricity. URL/DOI: www.entsoe.eu/data/power-stats/, https://doi.org/10.1109/TNNLS.2023.3259149.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhang, C.; Sjarif, N.N.A.; Ibrahim, R. Deep learning models for price forecasting of financial time series: A review of recent advancements: 2020–2022. *WIREs Data Min. Knowl. Discov.* **2024**, *14*, e1519. [CrossRef]
2. Sezer, O.B.; Gudelek, M.U.; Ozbayoglu, A.M. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Appl. Soft Comput.* **2020**, *90*, 106181. [CrossRef]
3. Ray, S.; Das, S.S.; Mishra, P.; Al Khatib, A.M.G. Time Series SARIMA Modelling and Forecasting of Monthly Rainfall and Temperature in the South Asian Countries. *Earth Syst. Environ.* **2021**, *5*, 531–546. [CrossRef]
4. Lai, Y.; Dzombak, D.A. Use of Integrated Global Climate Model Simulations and Statistical Time Series Forecasting to Project Regional Temperature and Precipitation. *J. Appl. Meteorol. Climatol.* **2021**, *60*, 695–710. [CrossRef]
5. Deb, C.; Zhang, F.; Yang, J.; Lee, S.E.; Shah, K.W. A review on time series forecasting techniques for building energy consumption. *Renew. Sustain. Energy Rev.* **2017**, *74*, 902–924. [CrossRef]
6. Van Zyl, C.; Ye, X.; Naidoo, R. Harnessing eXplainable artificial intelligence for feature selection in time series energy forecasting: A comparative analysis of Grad-CAM and SHAP. *Appl. Energy* **2024**, *353*, 122079. [CrossRef]
7. Rostami-Tabar, B.; Hyndman, R.J. Hierarchical Time Series Forecasting in Emergency Medical Services. *J. Serv. Res.* **2024.** [CrossRef]
8. Lim, S.S.Y.; Phouratsamay, S.L.; Yahouni, Z.; Gascard, E. Medicine consumption demand forecasting in French hospitals using Seasonal Auto-Regressive Integrated Moving Average (SARIMA) models. In Proceedings of the 2024 International Conference on Control, Automation and Diagnosis (ICCAD), Paris, France, 15–17 May 2024; pp. 1–6. [CrossRef]
9. Bergmeir, C.; Benítez, J.M. On the use of cross-validation for time series predictor evaluation. *Inf. Sci.* **2012**, *191*, 192–213. [CrossRef]
10. Ahmed, S.; Nielsen, I.E.; Tripathi, A.; Siddiqui, S.; Ramachandran, R.P.; Rasool, G. Transformers in Time-Series Analysis: A Tutorial. *Circuits Syst. Signal Process.* **2023**, *42*, 7433–7466. [CrossRef]
11. Wen, Q.; Zhou, T.; Zhang, C.; Chen, W.; Ma, Z.; Yan, J.; Sun, L. Transformers in Time Series: A Survey. *arXiv* **2023**, arXiv:2202.07125.
12. Liu, X.; Wang, W. Deep Time Series Forecasting Models: A Comprehensive Survey. *Mathematics* **2024**, *12*, 1504. [CrossRef]
13. Schwartz, R.; Dodge, J.; Smith, N.A.; Etzioni, O. Green AI. *Commun. ACM* **2020**, *63*, 54–63. [CrossRef]
14. Balderas, L.; Lastra, M.; Benítez, J.M. Optimizing dense feed-forward neural networks. *Neural Netw.* **2024**, *171*, 229–241. [CrossRef] [PubMed]
15. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2023**, arXiv:1706.03762.
16. Bolón-Canedo, V.; Morán-Fernández, L.; Cancela, B.; Alonso-Betanzos, A. A review of green artificial intelligence: Towards a more sustainable future. *Neurocomputing* **2024**, *599*, 128096. [CrossRef]
17. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. *arXiv* **2020**, arXiv:2005.14165.

18. Patterson, D.; Gonzalez, J.; Le, Q.; Liang, C.; Munguia, L.M.; Rothchild, D.; So, D.; Texier, M.; Dean, J. Carbon Emissions and Large Neural Network Training. *arXiv* **2020**, arXiv:2104.10350.
19. OpenAI. ChatGPT: A Large-Scale Generative Model for Open-Domain Chat. 2021. Available online: https://github.com/openai/gpt-3 (accessed on 5 April 2024).
20. Spadon, G.; Hong, S.; Brandoli, B.; Matwin, S.; Rodrigues, J.F., Jr.; Sun, J. Pay Attention to Evolution: Time Series Forecasting with Deep Graph-Evolution Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 5368–5384. [CrossRef]
21. Dong, E.; Du, H.; Gardner, L. An interactive web-based dashboard to track COVID-19 in real time. *Lancet Infect. Dis.* **2020**, *20*, 533–534. [CrossRef]
22. Silva, I.; Moody, G.; Scott, D.J.; Celi, L.A.; Mark, R.G. Predicting In-Hospital Mortality of ICU Patients: The PhysioNet/Computing in Cardiology Challenge 2012. *Comput. Cardiol.* **2012**, *39*, 245–248.
23. Le Guen, V.; Thome, N. Deep Time Series Forecasting with Shape and Temporal Criteria. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 342–355. [CrossRef] [PubMed]
24. Yuan, Y.; Kitani, K.M. Diverse Trajectory Forecasting with Determinantal Point Processes. *arXiv* **2020**, arXiv:1907.04967.
25. Salinas, D.; Flunkert, V.; Gasthaus, J.; Januschowski, T. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *Int. J. Forecast.* **2020**, *36*, 1181–1191. [CrossRef]
26. Oreshkin, B.N.; Carpov, D.; Chapados, N.; Bengio, Y. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv* **2020**, arXiv:1905.10437.
27. Thiede, L.; Brahma, P. Analyzing the Variety Loss in the Context of Probabilistic Trajectory Prediction. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9953–9962. [CrossRef]
28. Smyl, S.; Dudek, G.; Pelka, P. ES-dRNN: A Hybrid Exponential Smoothing and Dilated Recurrent Neural Network Model for Short-Term Load Forecasting. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *35*, 11346–11358. [CrossRef]
29. Zhang, S.; Luo, J.; Wang, S.; Liu, F. Oil price forecasting: A hybrid GRU neural network based on decomposition–reconstruction methods. *Expert Syst. Appl.* **2023**, *218*, 119617. [CrossRef]
30. Homepage—U.S. Energy Information Administration (EIA). Available online: https://www.eia.gov/ (accessed on 20 May 2024).
31. Power Statistics. Available online: https://www.entsoe.eu/ (accessed on 14 May 2024).
32. De Gooijer, J.G.; Hyndman, R.J. 25 years of time series forecasting. *Int. J. Forecast.* **2006**, *22*, 443–473. [CrossRef]
33. Hyndman, R.J.; Koehler, A.B. Another look at measures of forecast accuracy. *Int. J. Forecast.* **2006**, *22*, 679–688. [CrossRef]
34. Sakoe, H.; Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust. Speech Signal Process.* **1978**, *26*, 43–49. [CrossRef]
35. Frías-Paredes, L.; Mallor, F.; León, T.; Gastón-Romeo, M. Introducing the Temporal Distortion Index to perform a bidimensional analysis of renewable energy forecast. *Energy* **2016**, *94*, 180–194. [CrossRef]
36. Löning, M.; Bagnall, A.; Ganesh, S.; Kazakov, V.; Lines, J.; Király, F.J. sktime: A Unified Interface for Machine Learning with Time Series. *arXiv* **2019**, arXiv:1909.07872.
37. Király, F.; Löning, M.; Bagnall, T.; Middlehurst, M.; Ganesh, S.; Walter, M.; Oastler, G.; Ray, A.; Lines, J.; ViktorKaz; et al. sktime/sktime: v0.32.4. *ZENODO* **2024**. [CrossRef]
38. Dudek, G. Pattern similarity-based methods for short-term load forecasting—Part 2: Models. *Appl. Soft Comput.* **2015**, *36*, 422–441. [CrossRef]
39. Taylor, S.J.; Letham, B. Forecasting at Scale. *Am. Stat.* **2018**, *72*, 37–45. [CrossRef]
40. Dudek, G. Neural networks for pattern-based short-term load forecasting: A comparative study. *Neurocomputing* **2016**, *205*, 64–74. [CrossRef]
41. Pełka, P. Pattern-based Forecasting of Monthly Electricity Demand using Support Vector Machine. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–8. [CrossRef]
42. Pełka, P.; Dudek, G. Pattern-based Long Short-term Memory for Mid-term Electrical Load Forecasting. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 9–24 July 2020; pp. 1–8. [CrossRef]
43. Pełka, P.; Dudek, G. Neuro-Fuzzy System for Medium-Term Electric Energy Demand Forecasting. In *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology—ISAT 2017*; Springer: Cham, Switzerland, 2018; pp. 38–47.
44. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Chang, X.; Zhang, C. Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, 6–10 July 2020; pp. 753–763. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.