university of groningen / faculty of science and engineering

Transfer Learning for Short-Term-Load-Forecasting: Convolutional Neural Network - Long Short-Term Memory Forecasting Approach

Dan-Andrei Avram

University of Groningen

Transfer Learning for Short-Term-Load-Forecasting: Convolutional Neural Network - Long Short-Term Memory Forecasting Approach

Bachelor's Thesis

To fulfill the requirements for the degree of
Bachelor of Science in Computing Science
at University of Groningen under the supervision of
Dr D. Düştegör (Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen)
and
A. Tello Guerrero (Bernoulli Institute for Mathematics, Computer Science and Artificial Intelligence, University of Groningen)

Dan-Andrei Avram (s4127323)

February 28, 2024

Contents

Acknowledgments

I would like to express my immense gratitude to each person that has guided me and taught me leading up to this project and throughout its creation process.

Firstly, to my supervisors Dilek and Andrés, thank you profoundly for your help. Your knowledge in the domain, shared through invaluable insights and feedback when needed, guided the directions of this research towards a final product which I am honored to present, and would not have done as well if not for your patience.

I'm thankful for my family's and friends' immense love, support, and belief in me that have allowed me to reach a potential of which prospect I wouldn't have imagined just a few years ago. These people provided me with invaluable strength and vision and I hope sincerely that I can reciprocate this support in a much greater measure.

Lastly, I would like to extend my gratitude to the academics who have built and compounded all the innovation related to this domain of Computing Science and led me to discovering exactly what it means to stand on the shoulders of giants. This especially includes the academics of the Rijksuniversiteit Groningen; my higher education life here has revealed to me an innumerable amount of experiences and opportunities which will remain in my life forever.

Abstract

In contemporary energy systems, accurate short-term energy load forecasting is crucial from both economical and environmental standpoints, and Deep Learning has demonstrated its efficacy as a powerful tool for this purpose. However, in order for Deep Learning models to achieve optimal performance, it is typically necessary for them to undergo extensive training, using a substantial amount of data. Nevertheless, this is not possible to achieve for newly constructed buildings due to their very limited amount of historical data. The utilization of Transfer Learning (TL) is able to counteract this issue by applying the data of the target domain onto the learning of a model which has already been trained on external, more comprehensive data. While previous studies have demonstrated the high performance of hybrid models such as the Convolutional Neural Network - Long Short-Term Memory (CNN-LSTM) for energy load forecasting, and the benefits of TL in enhancing the performance of conventional LSTM models, this research work addresses the existing literature gap of the analysis of TL being performed on a CNN-LSTM model. Focusing on energy load forecasting for schools in New York, the findings of this study provide further evidence for the efficacy of TL on an LSTM model. However, CNN-LSTM yields inconsistent outcomes under the effects of TL. Ultimately, this latter model achieves its optimal performance in our setting not by means of TL, but rather through improved data pre-processing; the omission of weather data during the training process played an important role in achieving the best performing CNN-LSTM model, contrary to the initial expectations that a more complex training dataset would lead to better forecasting results.

1 Introduction

The implementation of accurate forecasting techniques facilitates energy management practices, and is hence holding significant importance environmentally and economically [1]. Machine Learning (ML) models such as Long Short-Term Memory (LSTM) networks, Convolutional Neural Networks (CNN), as well as combinations of ML models (forming a hybrid model) have recently become popular and promising approaches for energy load forecasting. These models have shown remarkable effectiveness in handling complex time series data and achieving high levels of prediction accuracy [2].

The application of ML and Deep Learning (DL) models is effective provided that significant training data is available [3]. New buildings do not have historical data to feed a model to predict their energy consumption. To counteract the problem of limited historical data, the literature reports Transfer Learning (TL) as a potential solution [4] [5] [6]. TL consists of training models on an existing source domain (e.g. a series of buildings with an adequate amount of historical data), then the pre-trained models are used on a target domain (i.e. new building, lacking historical data) [4].

This study is inspired by prior research that indicates the hybrid CNN-LSTM model is a very successful method for time series energy load forecasting, when given access to at least energy consumption and weather data [7]. Inspiration for this work also includes research which displays the capability of LSTM in a TL scenario [8].

Buildings within the education sector are distinctive for their variation in energy usage. Generally, in the span of 24 hours, most of the activity in a school (and implicit energy usage) takes place during the day, whereas during the nighttime the energy demand significantly decreases as a result of limited student activity. Vacations and weekends are also a defining factor in the energy load of this type of building, implying entire weeks of significantly lower energy usage. However, these patterns occur predictably throughout specific periods of the year and week.

The implemented and tested models in this study utilize a publicly available dataset consisting of 100 anonymous buildings located in the United States (out of which there are 25 schools) which displays their energy usage for the entirety of the year 2012 [9]. For each building used in this project's models, weather data for the year 2012 has been collected from the Meteostat API and included in the training of the models [10]. The examination of excluding weather data has also been investigated; further sections will showcase that this omission yielded different results depending on whether the model utilized was hybrid or non-hybrid.

Section 2 will present the State of the Art, concluding with the reasoning behind choosing a conventional LSTM model and a hybrid CNN-LSTM model. Section 3 will discuss the methodology of our paper, providing general explanations and definitions on the concepts relevant to our research. Section 4 will go in-depth into how these latter tools are integrated into a software framework of multiple ML models and evaluations. In Section 5, this framework ultimately leads to the results and implicit analysis of their values. Section 6 will discuss the insights derived from our research, as well as potential further work to broaden this study.

2 State of the Art

In the context of energy load forecasting, numerous instances of LSTM have demonstrated encouraging outcomes [11] [12]. The papers utilize LSTM, a type of recurrent neural network (RNN) architecture, for the purpose of modeling and forecasting power demand. LSTM networks are very suitable for energy usage forecasting due to their ability to effectively capture long-term dependencies, a characteristic frequently observed in time series data like energy load. Control gates and memory cells allow LSTM to regulate the flow of what data it remembers or forgets [13].

A paper by Eunjeong Choi et al. [13] further supports the efficacy of LSTM, implementing an effective power demand forecasting system using this RNN technique. The research gathered power demand data at a five-minute resolution from a variety of structures, such as public offices, residential areas, hospitals etc. This is relevant to our work, which will focus on 5-minute data, sampled to hourly data. The study of Choi ran tests with an emphasis on various types of forecasting: seasonal, long-term, and short-term. A comparison has been done between an LSTM model, a mixed data sampling method (also known as MIDAS, a statistical framework that enables the incorporation of variables sampled at varying frequencies), and the combination of the two. When compared to the MIDAS model, LSTM showcased lower error scores for each type of building it analysed, with root mean square error (RMSE) differences of at least 4.5 kWh, and as large as 23.3 kWh. This research not only provided additional evidence for the effectiveness of a standard LSTM model when used in energy forecasting for various building types, it also showed the power of the model when used in tangence with a model of another type, forming a hybrid model. While the latter model sometimes resulted in a RMSE slightly higher than the standard LSTM (0.3 kWh), it usually performed better, with errors lower than the LSTM performance, with differences in RMSE as large as 10.7 kWh. Similarly, our study will explore a hybrid model which combines LSTM, paired with a CNN.

A paper by Kim et al. [8] showcases the efficacy of using TL in an LSTM for forecasting energy usage in a building, which was a better tool for this task than the standalone LSTM. Transfer Learning is an ML approach involving the refinement of a pre-trained model on a similar, new problem; instead of using only the target domain for the purpose of optimizing a model, using additional data beforehand for this purpose can provide a better result, despite the possibility that this latter data has differences [14] [15]. Kim's study successfully showcased that the LSTM model performed better at hourly energy usage prediction when pre-trained on a source domain before training on the target domain, as opposed to not applying Transfer Learning. The RMSE differences between the TL-LSTM and the stand-alone LSTM were as high as 20% [8].

Chung and Jang [7] show the performance of LSTM in tandem with a CNN, forming a hybrid model. The CNN is able to iterate through the data input while performing convolution (which comes down to multiplication and summation of certain parts of the input) and pooling (which generally means averaging clusters of the convoluted data into more concise data), in such way that it is able to filter important features [16] [17]. [7] shows that a hybrid model consisting of a CNN and an LSTM is a very good approach when it comes to predicting energy usage, markedly outperforming a MLP model, a standard RNN model, and a stand-alone LSTM model.

The research paper [18] presents a model for the purpose of energy load forecasting, for an energy system that simultaneously operates multiple types of energy resources, resourced from a real-world dataset. The model performs Federated Learning as its training approach, which is a concept created by Google in 2017, referring to a decentralized methodology to overcoming data privacy related issues by aggregating model updates computed locally [19]. The model in question integrates the advantageous aspects of CNN and LSTM together, akin to the other hybrid models mentioned in this section, and additionally incorporates to it an attention mechanism. The layers of the proposed model fulfill distinct roles: the CNN is responsible for extracting overall features from the time series data, the attention mechanism assigns varying weights to distinct features according to their significance, and the LSTM captures data dependencies in the longer term. The findings indicated that the hybrid models exhibited superior precision compared to individual models, providing additional support for the efficacy of multiple models in synergy.

Nicolai Bo Vanting et al. [17] have run a comprehensive scoping review of the best-performing DL models in the subject of energy load forecasting. The paper discusses various applications in the energy domain, with contexts including single or multiple households, public or office buildings, larger districts such as countries or states. The study shows the absence of a universally flawless model, and the implicit need of the exploration of non-hybrid models before delving into more complex models. Nevertheless, the research report demonstrates that the CNN-LSTM model is generally a very promising candidate for energy load forecasting.

The existing body of literature on energy load forecasting has demonstrated the potential of LSTM and other DL techniques, especially the hybrid CNN-LSTM model. TL has demonstrated promising results as well. However, it is worth noting that there is a significant gap in the literature on the utilization of this hybrid model in a TL context. This aspect has particular significance for time series energy load forecasting in buildings with limited historical data. Consequently, the focus of this study will be the investigation of the following question: “To what extent can Transfer Learning increase the prediction performance of a Convolutional Neural Network Long Short-Term Memory model, when forecasting energy load for a school with limited historical data?”.

The subsequent sections will integrate some of the most effective methodologies explored in the field, resulting in a TL-CNN-LSTM model, which will be assessed in relation to the challenge of limited historical data availability. The architecture of this hybrid model will be inspired by the model created in [7]. In order to facilitate a comprehensive comparison, a TL-LSTM model will also be incorporated, for multiple reasons. Firstly, as also suggested by [17] it is important to first analyse a non-hybrid approach before relying on hybrid models. Secondly, the effectiveness of TL on a non-hybrid LSTM model has been demonstrated in Kim’s research [8]. Therefore, their architecture will be the inspiration for the development of our LSTM model. In order to assess whether the implementation of complex ML models is worth their implicit computation power in this scenario, our evaluations will also include a naive baseline model.

3 Methodology

3.1 Machine Learning and Deep Learning

The domain of energy load forecasting has witnessed a growing significance of ML and its more sophisticated variant, DL. ML is a subfield of Artificial Intelligence (AI) that facilitates the acquisition of knowledge by computer systems through the analysis of data. DL is a subset of ML which implies the utilization of neural networks to acquire representations of unprocessed input data. Using these techniques, distinct features can be derived from input data with high performance, without the need for explicit programming. These methods are well-suited for use with intricate, non-linear data, being able to accurately predict it, even if it contains possible seasonality or variability [20] [21].

3.2 Convolutional Neural Networks

CNN's are a family of DL algorithms that are typically employed for image classification tasks, working with two-dimensional data [22]. Nevertheless, they can also process a single dimension, which also makes them suitable for time series data, and hence able to perform energy load forecasting [23].

The CNN is able to iterate through the data input while performing convolution (which comes down to multiplication and summation of certain parts of the input) and pooling (which generally means averaging clusters of the convoluted data into more concise data), in such way that it is able to filter important features [17] [16].

3.3 Long Short Term Memory

LSTM is a particular kind of temporal cyclic neural network created to solve the long term dependency issue that affects most recurrent neural networks. LSTM employs memory units, which have a structure of an input gate, an output gate, and a forgetting gate, in place of the hidden layers of a conventional RNN. At each time step, these gates regulate the flow of what data keep or to remove. Filtering out less important information even over an extended period of time, LSTM has a great ability to "remember" pertinent long-term information of the dataset it works with, being able to perform well even in data exhibiting seasonality [24] [13].

3.4 Hybrid Machine Learning Models

Combining functionalities of multiple ML models, hybrid ML models have the potential effectively forecast energy load data, regardless of whether the data exhibits linear or nonlinear characteristics [25]. The utilization of various models allows for the exploitation of each of their respective strengths. In the context of a CNN-LSTM, the convolutional layers have the capability to filter important high-level features of the model's input data, while the incorporation of LSTM allows the model to effectively retain longer-term details of this data [16] [17].

3.5 Transfer Learning

TL enables ML models to achieve better forecasting or classifying results on limited data. This is done by incorporating knowledge gained from pre-training on external, potentially less relevant data than the target domain. Following this, the model will not need as much processing of the target domain to understand its low-level, universal information. Despite the latter dataset being possibly very scarce, or very different from the initial training dataset of the model, the model is now capable of forecasting or classifying this data better by applying the knowledge of relevant similarities from the training dataset. [4] [5] [8] [14]

4 Implementation

All of the steps of the implementation of these models have been done on the Google Colab platform. This includes the data loading, pre-processing including joining the energy usage data with the weather data of the building's anonymised location, data filtering, merging of multiple buildings data, normalizing. Training, testing, and plotting will also be done using this tool.

4.1 Dataset

The dataset provided by EnerNOC is a publicly accessible repository consisting of energy usage statistics for several commercial buildings in the U.S. The dataset covers this data for a total of 100 buildings of various types, for the year of 2012, in intervals of 5 minutes. The dataset also contains the location of the buildings, their size, type (Commercial properties, Education, Food Sales & Storage, Light Industrials), and subtype (Shopping Center, Bank, Primary/Secondary School etc.). Buildings are identified by integer id's of two or three digits.

For the usage of our ML models, we will exclusively consider buildings classified under the Education category. In this dataset, there are 25 such schools. The amount of schools initially given by the EnerNOC dataset is presented in Table 1. The amount of schools that will be used for our purposes is 10 New York schools, as described in the Data-Processing section of this paper.

Weather data may have a positive effect on the accuracy of our models. Therefore, this will be retrieved from the Meteostat API [10]. Meteostat is an openly available platform offering free access to historical meteorological and climate data, including the year 2012, which corresponds to the data available in the EnerNOC dataset. It indicates hourly weather data given a longitude and latitude, which are columns available for each building in the EnerNOC dataset.

The EnerNOC dataset, even when filtering out every building type except Education, presents diverse data variability per school. This variability can be seen in Figures 1 and 2. This may be attributed to noise in the data due to meter error, or it may actually indicate the genuine nature of the data for the specific school, where its energy load fluctuates more than other buildings. In this study, we will proceed on the assumption that the data provided is legitimate, and will therefore not attempt to alter the value of apparent outliers.

Table 1: Summary of the EnerNOC Dataset

US State	Amount of EnerNOC schools
New York	15
Chicago	7
Denver	3

4.2 Data Pre-processing

In raw form, the time series data per school in the EnerNOC dataset consists of the entire year 2012, in steps of 5 minutes, for which there are columns for timestamps, energy usage, and the timestamp

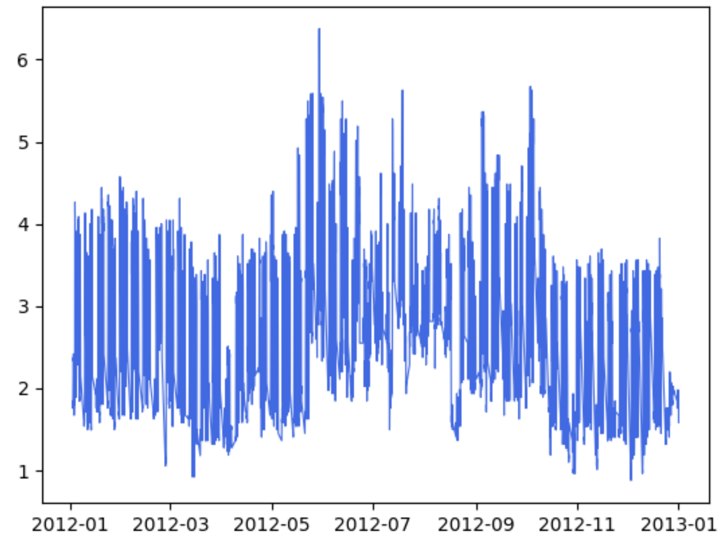

Figure 1: KWh values of school 92 throughout year 2012, presenting high data variability.

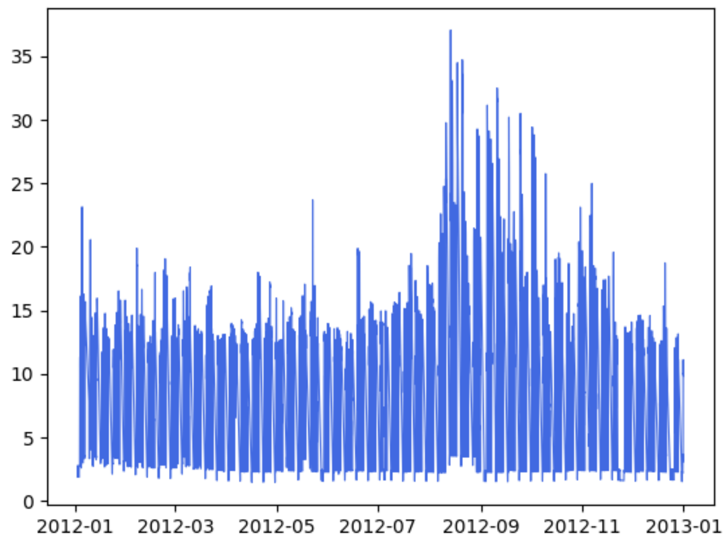

Figure 2: KWh values of school 136 throughout year 2012, presenting low data variability.

in UNIX format. The data also contains two almost consistently empty columns (named “estimated” and “anomaly”), which can be regarded as redundant, or even damaging to use in the training of the models, hence they were removed. Metadata is also present in a separate file, consisting of building size, latitude, longitude, state, building type (Commercial properties, Education, Food Sales & Storage, Light Industrials), and subtype (Shopping Center, Bank, Primary/Secondary School etc.) For every building in this project, its size has been concatenated to the dataset so that every row includes the metadata.

Considering data from multiple states, therefore possibly considering schools from very different regions, introduces an additional variable that may not be significant for addressing the research subject at hand. As this paper will mainly focus on the performance of certain models in a TL setting, we have restricted our analysis to schools in New York, as this state has the highest number of schools in the EnerNOC dataset that meet our aforementioned data filtering criteria.

Since there can be missing values in the data, the datasets have been chosen to be part of the training, validation, or testing sets only if they miss less than 10 rows in the EnerNOC dataset, or if they miss less than 1% of rows in the Meteostat dataset. For the schools that miss some data but have still been accepted due to respecting the aforementioned conditions, missing data has been linearly interpolated. Continuing the discussion on data imperfections, two duplicate datasets have been found and subsequently removed.

Out of the various information given by the library, the specific weather data that will be taken into consideration for this paper's training of models has been chosen based on the general findings on high correlation with the energy usage column of the EnerNOC schools. Two examples of Pearson correlation evaluations are shown in Figures 3 and 4. Based on which columns are never null for any school and generally have a high correlation with the energy usage of a building, the following columns given by the dataset have been included in our work: temp (temperature), dwpt (dewpoint), and rhum (humidity). As Meteostat's lowest interval of data is hourly, our data of 5 minute intervals collected from EnerNOC will be resampled to showcase hourly data.

Due to the nature of schools to be drastically less active during weekends, these periods have been dropped for every school. This decision was made because the increased data variability brought by the inclusion of weekends could potentially have a negative impact on the performance of the models. Therefore, only business days are taken into consideration.

Energy values during the night are also consistently low, and therefore possibly less important to keep track of than energy usage during the day, just like weekends. However, these nightly values have been kept, because of the possibility of the schools to have different amounts of active hours than other schools. For instance, one school may technically have 45 active hours per week, while another one may have more or less, making it difficult to choose a specific nightly period to exclude from the dataset.

The dataset includes time data in the form of timestamps and Unix values. This data has been modified into sinusoidal values, for the models to understand the cyclability of time. If the models work with time columns that are not properly engineered, they may not fully understand that the seventh day of the week is followed by the first one, or that in a day, the hour 23:00 is followed by 00:00. By modifying the time data into sinusoidal values for days, weeks, and months, cyclical time steps such as hours in a day, the day of the week, or the day of the month, are consistently far apart from each other.

The paper will also examine the validity of the argument presented in [17], stating that weather data is very important to include in training, for a successful model in energy load forecasting. Therefore, data processing will result in two different datasets, each analysed separately: one dataset will include hourly data for temperature, dewpoint, and humidity, while the other one will not.

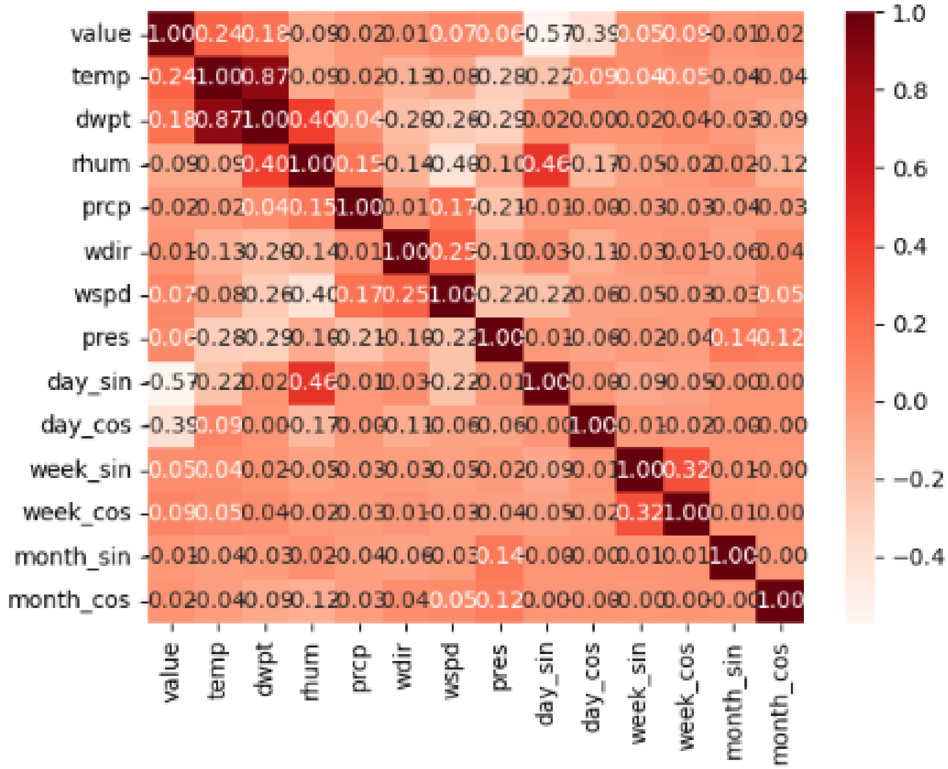

Figure 3: Pearson Correlation table for school 197. The column 'value' represents the kWh usage, the columns 'temp', 'dwpt', 'rhum', 'prcp', 'wdir', 'wspd', 'pres' are collected from Meteostat, while the columns 'day_sin', 'day_cos', 'month_sin', 'month_cos', 'year_sin', 'year_cos' are the cyclical sinusoidal values appended to each row based on timestamp.

The composition of the dataset, post-processing, is represented in Table 2, which enumerates the columns included in each data entry.

4.3 Simulating Transfer Learning Periods

Out of the list of schools chosen in the aforementioned data preprocessing step, one school identification number is randomly selected, then dropped from the list of schools used for the training or validation sets, and shall be used instead for TL. Our paper focuses on the efficacy of pre-trained models with TL on schools that lack historical data. Acknowledging that the EnerNOC dataset shows data for a whole year (with the exception of any missing data resulting from errors), we can simulate a novel school by choosing a very limited period of a year for the training and testing of TL. Hence, for the TL training set, we opt for a random 30 day period from the randomly chosen school, while for the TL test set we choose the 30 days directly following the train set period, thus simulating a school with only about 2 months of historical data. A representation of this process is shown in Figure 5.

4.4 Training and validation sets

After choosing the TL school and excluding it from the rest of the schools, two other school datasets are randomly chosen to be the validation dataset for each model of the paper. The remaining school

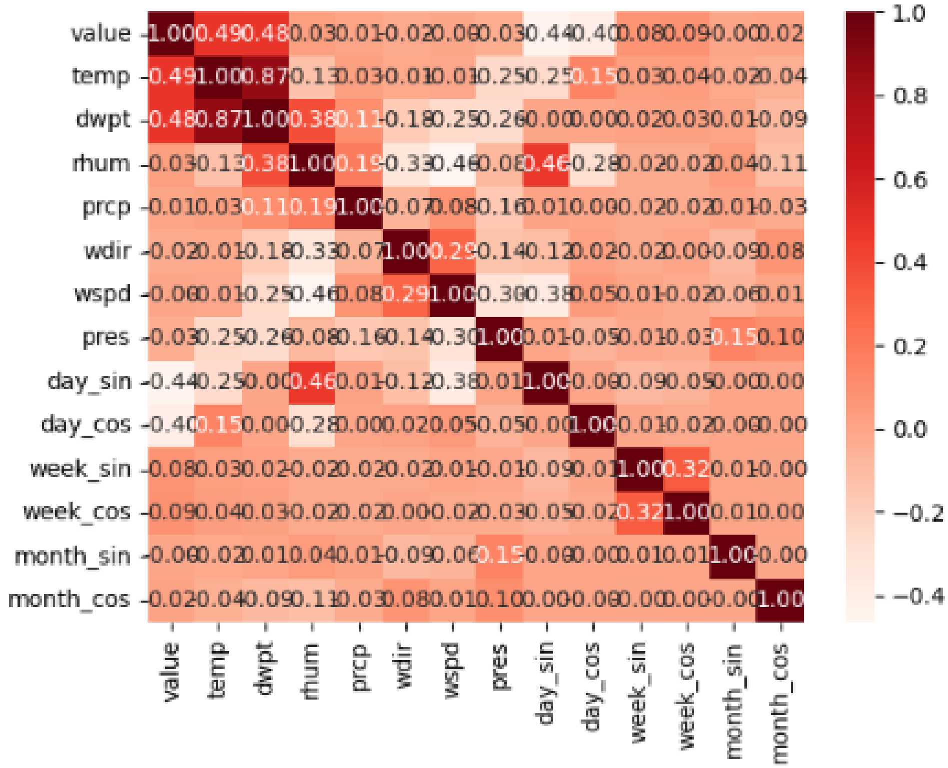

Figure 4: Pearson Correlation table for school 217. The column 'value' represents the kWh usage, the columns 'temp', 'dwpt', 'rhum', 'prcp', 'wdir', 'wspd', 'pres' are collected from Meteostat, while the columns 'day_sin', 'day_cos', 'month_sin', 'month_cos', 'year_sin', 'year_cos' are the cyclical sinusoidal values appended to each row based on timestamp.

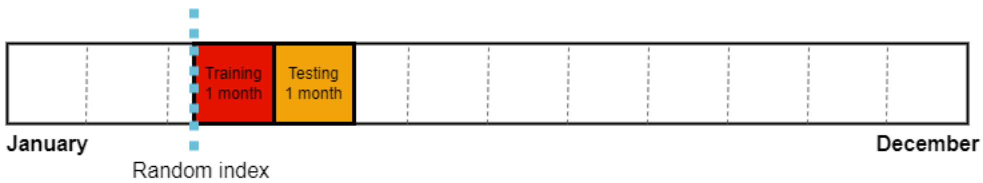

Figure 5: Representation of a school dataset being used for simulating a TL school. An index is randomly chosen, then a month following it is used for TL training, then the following month is used for TL testing. Note that the random index is not necessarily placed at the start of a month.

datasets will be used for the purpose of model training. It is important to note that the train set and validation set are based on the same one-year period: our separation into training and validating sets is only made by building identification number. It has been considered that the added data diversity in the validation set inferred by this approach may be beneficial for model performance, as the hyper-parameter tuning will focus on reducing the error of the models on the validation set.

Once the Train set and Validation set have been established, both for before and during TL, the data is subjected to normalization, being scaled down into a range from 0.0 to 1.0 using the MinMaxScaler feature of the scikit-learn Python library [26].

Table 2: Input Data Columns. Columns ‘temp’, ‘dwpt’, ‘rhum’ are not included in the datasets without weather.

Column Name	Description
value	Energy value
sq_ft	School size in square feet
state_label	State label, as a number
day_sin	Sinusoidal value for day, as sinus
day_cos	Sinusoidal value for day, as cosinus
month_sin	Sinusoidal value for month, as sinus
month_cos	Sinusoidal value for month, as cosinus
year_sin	Sinusoidal value for year, as sinus
year_cos	Sinusoidal value for year, as cosinus
temp	Temperature
dwpt	Dewpoint
rhum	Humidity

4.5 Windows of Features and Labels

The sizes that have been chosen for the input and output of each model are the following: the input represents 120 rows, each indicating data for one hour. Consequently, the data spans 5 days, containing all the columns determined in the data processing stage, which may or may not include weather data. Regardless of the inclusion of weather data in the dataset and the specific model employed, the output of one prediction will always be the same: 1 row by 1 column, representing the energy value directly following the 120 rows of the input. Ultimately, iterating through the windows of features and labels of one set will be done as follows: the model traverses through all the windows of a school in a chronological order, then proceeds to the next school in the set, until all schools of the set have been processed.

4.6 Machine Learning Models

Before delving into TL, this paper will first focus on the efficacy of two models on the EnerNOC dataset, with or without including weather data from the Meteostat library. We will also include a baseline model, to evaluate whether the added complexity and implicit needed computation power of a deep learning model [27] is worth it in comparison with implementing a much more naive model. Specifically, our baseline model receives the same input as the CNN-LSTM or LSTM models, but only outputs the same energy value as the latest energy value in the input, without any further calculations. While this is a naive approach, it creates a very fast and efficient model. Additionally, as it does not take any other column or row into consideration, this will make it result in the same values for RMSE and resepectively MAE, for the data with or without weather included.

Tables 3, 4, 5 summarise the models, while Tables 6, 7 go in more detail on the hyperparameter tuning of the layers of the LSTM and CNN-LSTM models. The number of columns in the input layer will always either be 9 or 12, depending on whether we choose the weather data (temperature, dewpoint, and humidity) to be included. All of these columns can be seen in Table 2.

Table 3: Base Model Layer Overview

Layer	Details	Optional?
Input Layer	120x9 if excl. weather 120x12 if incl. weather	No
Output Layer	1x1, energy value of the last input row	No

Table 4: CNN-LSTM Model Layer Overview

Layer	Details	Optional?
Input Layer	120 x 9 if excl. weather 120 x 12 if incl. weather	No
Conv1D (1st layer)	Size determined during hyperparam. tuning	No
Conv1D (2nd layer)	Size determined during hyperparam. tuning	No
Conv1D (3rd layer)	Size determined during hyperparam. tuning	Yes
Dropout (1st layer)	Size determined during hyperparam. tuning	No
Conv1D (4th layer)	Size determined during hyperparam. tuning	Yes
Pooling	Type determined during hyperparam. tuning	No
LSTM (1st layer)	Size determined during hyperparam. tuning	Yes
Dropout (2nd layer)	Size determined during hyperparam. tuning	No, if LSTM Layer above exists
LSTM (2nd layer)	Size determined during hyperparam. tuning	No
Dropout (3rd layer)	Size determined during hyperparam. tuning	No
Dense	1x1, the expected next energy value	No

Table 5: LSTM Model Layer Overview

Layer	Details	Optional?
Input Layer	120 x 9 if excl. weather 120 x 12 if incl. weather	No
LSTM (1st layer)	Size determined during hyperparam. tuning	No
Dropout (1st layer)	Size determined during hyperparam. tuning	No
LSTM (2nd layer)	Size determined during hyperparam. tuning	No
Dropout (2nd layer)	Size determined during hyperparam. tuning	No
LSTM (3rd layer)	Size determined during hyperparam. tuning	Yes
Dropout (3rd layer)	Size determined during hyperparam. tuning	No, if LSTM Layer above exists
LSTM (4th layer)	Size determined during hyperparam. tuning	Yes
Dropout (4th layer)	Size determined during hyperparam. tuning	No, if LSTM Layer above exists
LSTM (5th layer)	Size determined during hyperparam. tuning	No
Dropout (5th layer)	Size determined during hyperparam. tuning	No
Dense	1x1, the expected next energy value	No

4.7 Hyperparameter Tuning

To reach optimal model performance, Keras Tuner has been used for performing hyperparameter tuning. This allowed us to vary parameters such as the learning rate of a model, as well as the amount of layers, and the parameters used in of each of the layers.

Whether or not weather is included in a model's training, the model's possible hyperparameter ranges look the same, the only difference being the amount of columns that it receives in the input layer: 9 or 12. Individual hyperparameter tuning is still performed for each model, with or without weather data; for instance, the best CNN-LSTM without weather may include a single convolutional layer, while including the data may need the model to have an additional convolutional layer.

The CNN-LSTM model was inspired by the architecture mentioned in [7]. Our model therefore starts off with two layers with filter sizes as large as 128. There is an added possibility for more than two convolutional layers. Similarly to Chung's approach, dropout is being done, and after all the convolutional layers, pooling is performed, followed by an LSTM layer with units up to 128. The hyperparameter ranges for the CNN-LSTM model can be seen in Table 6.

Table 6: CNN-LSTM Hyperparameter Ranges

Hyperparameter	Range	Step Size	Notes
Learning Rate	[0.00001, 0.01]	logarithmic	
Batch Size	64		Fixed Value
Filters for CNN Layer 1	[32, 128]	32	
Kernel size for CNN Layer 1	[1, 3]	1	
Filters for CNN Layer 2	[32, 128]	32	
Kernel size for CNN Layer 2	[1, 3]	1	
Include CNN Layer 3?	[0, 1]	1	0=No, 1=Yes
Filters for CNN Layer 3	[32, 128]	32	
Kernel size for CNN Layer 3	[1, 3]	1	
Dropout Layer 1 Rate	[0.0, 0.5]	0.25	
Include CNN Layer 4?	[0, 1]	1	0=No, 1=Yes
Filters for CNN Layer 4	[32, 128]	32	
Kernel size for CNN Layer 4	[1, 3]	1	
Pooling Type	[Avg, Max]		Categorical Choice
Include LSTM Layer 1 and Dropout 2?	[0, 1]	1	0=No, 1=Yes
Units for LSTM Layer 1	[16, 48]	16	
Dropout Layer 2 Rate	[0.0, 0.5]	0.5	
Units for LSTM Layer 2	[32, 128]	32	
Dropout Layer 3 Rate	[0.0, 0.5]	0.25	

The primary source of inspiration for our LSTM model's architecture was detailed in [8]. Similarly to their architecture, the amount of layers included in the model was up to 5, and the amount of units of each layer increased the higher in the model the layer was. Hence, the possible unit amounts are as low as 4 units in the lower levels of the architecture, and as high as 96 in higher levels. The hyperparameter ranges for the LSTM model can be seen in Table 7.

Table 7: LSTM Hyperparameter Ranges

Hyperparameter	Range	Step Size	Notes
Learning Rate	[0.00001, 0.01]	logarithmic	
Batch Size	64		Fixed Value
Units for LSTM Layer 1	[4, 12]	4	
Dropout Layer 1 Rate	[0.0, 0.5]	0.25	
Units for LSTM Layer 2	[8, 24]	8	
Dropout Layer 2 Rate	[0.0, 0.5]	0.5	
Include LSTM Layer 3 and Dropout 3?	[0, 1]	1	0=No, 1=Yes
Units for LSTM Layer 3	[16, 48]	16	
Dropout Layer 3 Rate	[0.0, 0.5]	0.5	
Include LSTM Layer 4 and Dropout 4?	[0, 1]	1	0=No, 1=Yes
Units for LSTM Layer 4	[32, 96]	16	
Dropout Layer 4 Rate	[0.0, 0.5]	0.25	
Units for LSTM Layer 5	[32, 96]	16	
Dropout Layer 5 Rate	[0.0, 0.5]	0.5	

Each model has had 3 different hypertuning trials, and there have been two executions done per trial. At each model's compiling, the chosen optimizer was Adam [28], the loss was mean squared error, and the metric was mean absolute error. At the running of the tuning, the objective was the validation loss, meaning the mean squared error of the validation set. Additionally, each model's fitting has been done with 100 epochs, with an early stopping possibility at the passing of 20 epochs without an increase of performance on the validation loss.

The baseline model does not need hypertuning, as it is has the exact value of the latest energy value given by its input, while not having a correlation with any other row or column.

4.8 Best Hyperparameter Architectures

After performing hyperparameter tuning, we will have obtained 5 different models to evaluate: the baseline model, the CNN-LSTM's with and without weather data, and the LSTM's, likewise with and without weather data. The final architectures of each of these models are shown in Tables 8 and 10 for the dataset that includes weather, and respectively Tables 9 and 11 for the dataset that does not include weather.

Table 8: Best Hyperparameters of CNN-LSTM with Weather

Hyperparameter	Value
Learning Rate	0.000287
Filters for CNN Layer 1	96
Kernel Size for CNN Layer 1	3
Filters for CNN Layer 2	32
Kernel Size for CNN Layer 2	3
Include CNN Layer 3?	0 (No)
Dropout Layer 1 Rate	0.5
Include CNN Layer 4?	0 (No)
Pooling Type	“avg”
Include LSTM Layer 1 and Dropout 2?	0 (No)
Units for LSTM Layer 2	64
Dropout Layer 3 Rate	0.0

Table 9: Best Hyperparameters of CNN-LSTM without Weather

Hyperparameter	Value
Learning Rate	0.000645
Filters for CNN Layer 1	32
Kernel Size for CNN Layer 1	1
Filters for CNN Layer 2	64
Kernel Size for CNN Layer 2	3
Include CNN Layer 3?	1 (Yes)
Dropout Layer 1 Rate	0.25
Include CNN Layer 4?	0 (No)
Pooling Type	“max”
Include LSTM Layer 1 and Dropout 2?	0 (No)
Units for LSTM Layer 2	128
Dropout Layer 3 Rate	0.0
Units for LSTM Layer 1	48
Dropout Layer 2 Rate	0.0
Filters for CNN Layer 3	32
Kernel Size for CNN Layer 3	1

4.9 Hyperparameter Tuning Performance

After performing hyperparameter tuning, the train set and the validation set have been concatenated, and all the models have been retrained on this concatenated set, to develop knowledge that comprises more schools. Finally, each of the models with the best architecture trains on this concatenated set, and is evaluated in terms of RMSE and MAE. These error values can be seen in Table 12.

Table 10: Best Hyperparameters of LSTM with Weather

Hyperparameter	Value
Learning Rate	0.004902
Units for LSTM Layer 1	4
Dropout Layer 1 Rate	0.0
Units for LSTM Layer 2	24
Dropout Layer 2 Rate	0.0
Include LSTM Layer 3	1 (Yes)
Units for LSTM Layer 3	16
Dropout Layer 3 Rate	0.0
Include LSTM Layer 4	1 (Yes)
Units for LSTM Layer 4	80
Dropout Layer 4 Rate	0.5
Units for LSTM Layer 5	64
Dropout Layer 5 Rate	0.5

Table 11: Best Hyperparameters of LSTM without Weather

Hyperparameter	Value
Learning Rate	0.000427
Units for LSTM Layer 1	4
Dropout Layer 1 Rate	0.0
Units for LSTM Layer 2	8
Dropout Layer 2 Rate	0.5
Include LSTM Layer 3	1 (Yes)
Units for LSTM Layer 3	48
Dropout Layer 3 Rate	0.0
Include LSTM Layer 4	0 (No)
Units for LSTM Layer 4	48
Dropout Layer 4 Rate	0.0
Units for LSTM Layer 5	80
Dropout Layer 5 Rate	0.0

4.10 Transfer Learning Methods

We can perform many types of TL for a Neural Network, but the general approach is to train only some layers of the network (generally layers at a higher level), while retaining the original weights and biases of the rest (generally called “freezing” the layers) [29] [30].

Table 12: Performance Comparison of Models on data with weather and without weather

Error	Dataset	Metrics		
		Baseline	CNN-LSTM	LSTM
RMSE	With weather	1.59	1.32	**1.07**
	Without weather	1.59	**1.04**	4.08
MAE	With weather	0.98	0.97	**0.74**
	Without weather	0.98	**0.77**	3.06

For the purpose of this paper, we have decided to analyse multiple TL approaches, including two in which freezing layers is not performed. These methods are defined in Table 13. We will see, in later sections of this research, that not freezing layers is actually the best TL approach for our setting.

Table 13: Transfer Learning Methods Analysed

Transfer Learning Method	Description
Maximum Freezing	Only the last dense layer of the model is trainable. The LR used is the best LR found during hyperparameter tuning
Minimum Freezing 1	All layers of the model are trainable. The LR is 0.00001
Minimum Freezing 2	All layers of the model are trainable. The LR is the best LR found during hyperparameter tuning

4.11 Transfer Learning Experiment Results

The results of performing TL on the CNN-LSTM model and the best LSTM model can be seen in the Tables 14 and 15 and will be discussed in the following section. Note that TL, like the tuning of the hyperparameters, has no effect on the baseline model, and therefore the performance of TL will not be evaluated for it. Therefore, for the baseline model, the RMSE and MAE will remain the same as in Table 12.

Table 14: Performance Metrics for TL on the CNN-LSTM Model

CNN-LSTM	With weather		Without weather	
	RMSE	MAE	RMSE	MAE
Before TL	1.32	0.97	**1.04**	**0.77**
After TL with Minimum Freezing	1.54	1.19	1.94	1.46
After TL with Maximum Freezing 1	1.15	0.86	1.05	0.79
After TL with Maximum Freezing 2	1.76	1.31	1.46	1.13

Table 15: Performance Metrics for TL on the LSTM Model

LSTM	With weather		Without weather	
	RMSE	MAE	RMSE	MAE
Before TL	1.07	0.74	4.08	3.06
After TL with Maximum Freezing	1.04	0.77	3.98	2.98
After TL with Minimum Freezing 1	**0.96**	**0.68**	4.24	3.26
After TL with Minimum Freezing 2	1.58	1.04	4.08	3.16

5 Discussion

Compared with a baseline model, the best-performing CNN-LSTM and LSTM have always exhibited lower values for both the MAE and RMSE. This indicates that in terms of results, utilizing DL is a superior alternative over employing a naive forecasting model. If taking computational power into consideration for choosing the right model, the conclusion is more subjective, as will be explained later in this section.

Tuning the hyperparameters for the hybrid CNN-LSTM model concluded to provide results which were better than those of the tuned LSTM model, despite the fact that for the latter, hyperparameter tuning has been performed as well, and both models having a decent complexity. This further adds to the credibility of [17].

The paper aforementioned also suggested that weather data is part of the bare minimum data to include for a successful model. However, in the case of analysing the EnerNOC dataset, merged with weather data from Meteostat, weather data seems to be not only useless, but damages the performance of the model, in the case of the CNN-LSTM model. This could mean that for this model, the time of the year/day itself is enough to understand the energy usage of the school.

While the best CNN-LSTM model needed no weather to perform best, The LSTM model clearly demonstrated its best performance when incorporating weather data, as opposed to excluding it. This resulted in a very close resemblance of error rates for both of the models.

The premise of our paper was evaluating the performance of TL on an energy load forecasting model, given a school which has a very limited amount of historical data. Using the data inferred by the CNN-LSTM, the results in terms of TL seem to be inconclusive. Looking at the best performing CNN-LSTM (without weather data), TL has always decreased the performance of the model in the end, for every kind of TL, for both RMSE and MAE metrics. The only case in which TL helped was when weather data was included in the CNN-LSTM training, decreasing both the RMSE and the MAE, when performing TL with minimal freezing (maintaining the trainability of each layer), with a learning rate of 1e-5. However, this added performance is still worse than the CNN-LSTM trained on data without weather, before TL. An additional point against TL in this case is the fact that these differences after performing it are fairly small: the error metrics values before and after this TL were 1.32 and 1.15 for the RMSE, and respectively 0.97 and 0.86 for the MAE value. Since TL can be expensive computationally [27], it is arguable that the marginal performance improvement achieved by employing this approach may not justify the incurred cost. However, this increased performance does show that it may still be worthwhile to do TL in certain scenarios: if the error rates before learning are less optimal than this paper's, or when the selection of variables included in the TL process is more meticulous.

The lack of definitive findings can also be attributed to the restricted scope of our TL testing, which was limited to a single month from a single school. Our experiments can be replicated and assessed across several educational buildings. By determining the average outcome of these trials, a more certain conclusion can be drawn on the most effective technique.

In agreement with the findings presented in [8], TL increased the performance of our LSTM model. The TL type responsible for this result was TL with no freezing, with a learning rate of 0.00001, on school data which incorporates weather information. This approach yielded the only instance in which TL set new best results in our research, not only compared to the performance of LSTM prior to TL, but also surpassing the results of the highest-performing CNN-LSTM model. It is worth remembering that the aforementioned type of TL is also the only one that provided any positive results for the CNN-LSTM. While it is arguable whether these small increases in performance are worth the computational costs of TL, these results consistently showcase the potential of this type of TL.

The foundational, low-level layers of the ML models, were expected to not need changes during TL. However, it is noticeable that the trainability of all layers was necessary for TL to be successful at all. This suggests that the information captured in the first layers of the models has represented higher-level features than expected, or that the TL school differed significantly from the training data, requiring the model to change some fundamental expectations before being able to perform better. We suspect that the latter case has a lower chance of being true than the former, because the TL building, as well as all the other buildings used in our work, were of the same type, showing similar energy usage trends throughout days and nights.

When it comes to performing TL, sources generally provide variations of tlmax as the general practice [31] [32]. The findings of our study suggest not to underestimate the approach of keeping the trainability of the majority of the model's layers.

It's also worth noting one significant difference between the best-performing CNN-LSTM model and LSTM model, which is the necessity of weather data. The best results of the CNN-LSTM show when not including weather data, but the LSTM model exhibits noticeable deficiencies under the same conditions, while performing similarly to the hybrid model when it is able to process weather data as well. This demonstrates significant promise for utilizing the CNN-LSTM model in situations when meteorological data is unavailable.

6 Conclusion

As seen in the results above, the analysis of our paper's premise shows interesting, varied results. Based on the literature on which we primarily focused, certain ideas mentioned in these papers showed contrasting outcomes in relation to expectations within our specific setting. There was even a situation in which two different stances of one paper were analysed, where one was confirmed to be correct, and the other showed the contrary outcome compared to the original paper's conclusions, in the context of our paper.

As mentioned in the previous section, the final results of performing TL show that, to perform the best forecasting on a school with limited data, we can choose between a CNN-LSTM trained on data without weather, without performing TL, or an LSTM trained on data with weather, after performing TL with all the layers trainable, with a very low learning rate. This choice would ultimately boil down to the user's preferences, available data, and available computation power.

As also seen in the previous section, the best performing TL method was different from what's generally suggested in the literature as a prime example of TL.

6.1 Future Work

Further tuning the hyperparameters of the models can prove to be an effective approach to increasing their accuracy. While a model's architecture can prove to be the best for a specific dataset, the impact of its parameters can be very different for another dataset.

As an ML model's efficacy is dependent on the quality of its training data, there can also be done a revision on the preprocessing of the dataset, implying the removal of more outliers, even if this means removing entire months or seasons from certain schools in the dataset.

Certain schools showcase unpredictable data, not following the normal energy usage change in a building. Force majeure is a very powerful factor in the energy usage of schools, and may have been the cause for irregularities in some parts of the data. The chance of such events happening on a specific day, as well as how the administration of a particular school chooses to react to the event, are variables vastly different and more volatile than the data used in this paper. They surpass the models' prediction capabilities, and if data affected by these events is part of the model's learning process, it could be as damaging to the accuracy as noisy data. Particularly to New York, the state containing the schools analysed in this paper, the following events may have affected the data in an unforeseeable way: Hurricane Sandy [33], heatwaves in multiple areas of the U.S.A [34] etc. Researching such events further and their implications could help create better models.

Additional forms of TL can be experimented with. Initially, our research has limited the trainability of layers during TL to either a minimal or maximal extent. Additional experiments can be conducted to explore a wider range of frozen layer amounts. One additional aspect to consider while doing TL experiments is the learning rate of the model. In the aforementioned trials, it would be best to additionally apply a variety of learning rates: taking into account the findings obtained in our specific scenario, one may achieve improved outcomes by utilizing lower ends of possible learning rates.

The partitioning of the data into a training set and a validation set based on school id's yielded acceptable results. However, for future investigations, an alternative approach could involve temporal splitting of the data rather than school-based splitting. This would entail selecting a specific time-frame from all non-TL schools to be included in the training set, and a period directly following the training set to be used for validation purposes.

Bibliography

[1] G. Hafeez, K. S. Alimgeer, Z. Wadud, I. Khan, M. Usman, A. B. Qazi, and F. A. Khan, "An Innovative Optimization Strategy for Efficient Energy Management With Day-Ahead Demand Response Signal and Energy Consumption Forecasting in Smart Grid Using Artificial Neural Network," vol. 8, pp. 84415–84433.

[2] M. Cordeiro-Costas, D. Villanueva, P. Eguía-Oller, M. Martínez-Comesaña, and S. Ramos, "Load Forecasting with Machine Learning and Deep Learning Methods," vol. 13, no. 13, p. 7933.

[3] Y. Roh, G. Heo, and S. E. Whang, "A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective," vol. 33, no. 4, pp. 1328–1347.

[4] C. Li, G. Li, K. Wang, and B. Han, "A multi-energy load forecasting method based on parallel architecture CNN-GRU and transfer learning for data deficient integrated energy systems," vol. 259, p. 124967.

[5] A. Hosna, E. Merry, J. Gyalmo, Z. Alom, Z. Aung, and M. A. Azim, "Transfer learning: A friendly introduction," vol. 9, no. 1, p. 102.

[6] K. Wang, H. Yu, G. Song, J. Xu, J. Li, and P. Li, "Adaptive forecasting of diverse electrical and heating loads in community integrated energy system based on deep transfer learning," vol. 10.

[7] J. Chung and B. Jang, "Accurate prediction of electricity consumption using a hybrid CNN-LSTM model based on multivariable data," vol. 17, no. 11, p. e0278071.

[8] D. Kim, Y. Lee, K. Chin, P. J. Mago, H. Cho, and J. Zhang, "Implementation of a Long Short-Term Memory Transfer Learning (LSTM-TL)-Based Data-Driven Model for Building Energy Demand Forecasting," vol. 15, no. 3, p. 2340.

[9] "EnerNOC 100 Building Open Dataset IPython Notebook Analysis."

[10] C. S. Lamprecht, "Meteostat Developers."

[11] J. Bedi and D. Toshniwal, "Deep learning framework to forecast electricity demand," vol. 238, pp. 1312–1326.

[12] F. He, J. Zhou, Z.-k. Feng, G. Liu, and Y. Yang, "A hybrid short-term load forecasting model based on variational mode decomposition and long short-term memory networks considering relevant factors with Bayesian optimization algorithm," vol. 237, pp. 103–116.

[13] E. Choi, S. Cho, and D. K. Kim, "Power Demand Forecasting using Long Short-Term Memory (LSTM) Deep-Learning Model for Monitoring Energy Sustainability," vol. 12, no. 3, p. 1109.

[14] H. Park, D. Y. Park, B. Noh, and S. Chang, "Stacking deep transfer learning for short-term cross building energy prediction with different seasonality and occupant schedule," vol. 218, p. 109060.

[15] L. Wang, H. Liu, Z. Pan, D. Fan, C. Zhou, and Z. Wang, "Long Short-Term Memory Neural Network with Transfer Learning and Ensemble Learning for Remaining Useful Life Prediction," vol. 22, no. 15, p. 5744.

[16] H. Widiputra, A. Mailangkay, and E. Gautama, “Multivariate CNN-LSTM Model for Multiple Parallel Financial Time-Series Prediction,” vol. 2021, p. 9903518.

[17] N. B. Vanting, Z. Ma, and B. N. r. Jø rgensen, “A scoping review of deep neural networks for electric load forecasting,” vol. 4, no. 2, p. 49.

[18] H. Luo, H. Zhang, and J. Wang, “Ensemble power load forecasting based on competitive-inhibition selection strategy and deep learning,” vol. 51, p. 101940.

[19] B. McMahan, E. Moore, D. Ramage, S. Hampson, and p. u. family=Arcas, given=Blaise Aguera, “Communication-efficient learning of deep networks from decentralized data,” in *Artificial Intelligence and Statistics*, pp. 1273–1282, PMLR.

[20] B. Mahesh, “Machine learning algorithms-a review,” vol. 9, no. 1, pp. 381–386.

[21] M. Khalil, A. S. McGough, Z. Pourmirza, M. Pazhoohesh, and S. Walker, “Machine Learning, Deep Learning and Statistical Analysis for forecasting building energy consumption — A systematic review,” vol. 115, p. 105287.

[22] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, “CNN-RNN: A unified framework for multi-label image classification,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2285–2294.

[23] S. Akhtar, S. Shahzad, A. Zaheer, H. S. Ullah, H. Kilic, R. Gono, M. Jasiński, and Z. Leonowicz, “Short-term load forecasting models: A review of challenges, progress, and the road ahead,” vol. 16, no. 4060.

[24] Y. Wang, N. Zhang, and X. Chen, “A short-term residential load forecasting model based on LSTM recurrent neural network considering weather features,” vol. 14, no. 2737.

[25] N.-T. Ngo, A.-D. Pham, T. T. H. Truong, N.-S. Truong, and N.-T. Huynh, “Developing a hybrid time-series artificial intelligence model to forecast energy use in buildings,” vol. 12, no. 1, p. 15775.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” vol. 12, pp. 2825–2830.

[27] R. Ye and Q. Dai, “Implementing transfer learning across different datasets for time series forecasting,” vol. 109, p. 107617.

[28] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization.”

[29] N. Ullah, J. A. Khan, M. S. Khan, W. Khan, I. Hassan, M. Obayya, N. Negm, and A. S. Salama, “An effective approach to detect and identify brain tumors using transfer learning,” vol. 12, no. 5645.

[30] F. Mattioli, C. Porcaro, and G. Baldassarre, “A 1D CNN for high accuracy classification and transfer learning in motor imagery EEG-based brain-computer interface,” vol. 18, no. 6, p. 066053.

[31] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, “Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning,” vol. 35, no. 5, pp. 1285–1298.

[32] M. M. Islam, M. B. Hossain, M. N. Akhtar, M. A. Moni, and K. F. Hasan, “CNN based on transfer learning models using data augmentation and transformation for detection of concrete crack,” vol. 15, no. 287.

[33] g.-i. family=US Department of Commerce, given=NOAA, “Hurricane Sandy.”

[34] “July 2012 National Climate Report | National Centers for Environmental Information (NCEI).”