ACHILLEAS O. ACHILLEOS

Modeling and Optimizing of an Island's Energy System

SCHOOL OF MECHANICAL ENGINEERING

Section: Mechanical Design and Automatic Control

Supervisor: Prof. Ioannis Antoniadis

Athens 2020

--- κενή σελίδα ---

<u>Υπεύθυνη δήλωση</u> για λογοκλοπή και για κλοπή πνευματικής ιδιοκτησίας:

Έχω διαβάσει και κατανοήσει τους κανόνες για τη λογοκλοπή και τον τρόπο σωστής αναφοράς των πηγών που περιέχονται στον οδηγό συγγραφής Διπλωματικών Εργασιών. Δηλώνω ότι, από όσα γνωρίζω, το περιεχόμενο της παρούσας Διπλωματικής Εργασίας είναι προϊόν δικής μου εργασίας και υπάρχουν αναφορές σε όλες τις πηγές που χρησιμοποίησα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτή τη Διπλωματική εργασία είναι του συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις της Σχολής Μηχανολόγων Μηχανικών ή του Εθνικού Μετσόβιου Πολυτεχνείου.

Αχιλλέως Αχιλλέας

Περιεχόμενα

1. Introduction 8
1.1 Background and motivation 8
1.2 Smart Grid Overview 9
1.3 Aim of this study 10
2. Backround 11
2.1 Machine Learning 11
2.1.1 Learning Problems 12
2.1.2 Under- and Overfitting 13
2.1.3 Artificial Neural Network 14
2.1.4 Feedforward Neural Networks 15
2.1.5 Simple Neuron and Activation function 16
2.1.6 Regularization 18
2.1.7 Recurrent Neural Networks 20
2.1.8 Convolutional Neural Networks 22
2.2 Related Work 24
2.2.1 An Evaluation of Convolutional and Recurrent Neural Networks 24
2.2.2 Time series forecasting: ARIMA vs. Deep Learning 24
3. General informations 26
3.1 Dataset 26
3.2 System Specifications 26
3.2.1 TensorFlow 27
3.2.2 Keras 27
3.2.3 Pandas 28
3.3 Photovoltaic system and parameters 28
4. Forecasting 31
4.1 Input Values and Training Description 31
4.1.1 Total Load Forecast 31
4.1.2 Solar Energy Forecast 35
4.1.3 Wind Energy Forecast 38
4.1.4 Thermal Energy Forecast 41
4.2 Performance Evaluation Metrics 43
4.2.1 Models Comparison and selection 44
5. Optimization Algorithm 48
5.1 General Problem 48
5.2 Wind Energy problem description 50
5.3 Wind Energy algorithm 51
5.3.2 Determining battery size 52
5.3.3 Proof of solution viability 54
5.4 Peak Shaving Algorithm Description 55
5.4.1 Winter Period Example 57

5.4.2 Summer Period Example 60
5.5 Final Results 61
5.6 Model of the Islandic Power Plant....63
6. Conclusions and Future Work....65
6.1 Conclusions 65
6.2 Future Work....65
7. Tables....67
8. Figures....68
9. List of References....70
10. Greek Extended Summary 74
10.1 Εισαγωγή – Περιγραφή του προβλήματος 74
11. Ανάλυση Μεθόδου 76
11.1 Θεωρία Βαθιάς Μάθησης 76
11.1.1 Επιτηρούμενη και Μη-Επιτηρούμενη Μάθηση 76
11.1.2 Τεχνητός Νευρώνας....77
11.1.3 Συνάρτηση Ενεργοποίησης 77
11.1.4 Μοντέλα Νευρωνικών Δικτύων....78
12. Γενικές Πληροφορίες....79
12.1 Δεδομένα 79
12.2 Λεπτομέρειες Συστήματος 79
13. Διαδικασία Πρόβλεψης 80
14. Αλγόριθμος Βελτιστοποίησης 84
14.1 Αλγόριθμος Αιολικής Ενέργειας....84
14.2 Αλγόριθμος Peak Shaving....86
15. Σύνοψη – Συμπεράσματα....87

Abstract

The ever expanding of greenhouse gas emission and limitation of fossil energy sources are driving demand for the green energies. Among the variety of the renewable energy sources, the wind power and PV in large scale is known as the best replacement for the conventional source of energies. In this study, two smart algorithms have been built to manage an islandic power system. The algorithms are based in the idea of battery energy storage system (BESS). At first, a forecasting of the productions took place, as it was necessary to build the predictive management algorithms. The prediction algorithm was built in Python. The forecasting used Deep learning methods and especially, Convolutional Neural Networks and Recurrent Neural Networks. In case of wind energy, a new control strategy is presented to manage the amount of energy that is generated by wind farm plant. As the battery plays a fundamental role in the control system. The control method reduces the fluctuation of supplied wind power while it empowers the operator to make a balance between energy supply and demand in a profitable way using battery energy storage. After that, a predictive energy management (EMS) algorithm was developed, capable of load smoothing and peak shaving of the maximum demand values, simultaneously. In this way, the maximum demand of the island's system, was covered from stored renewable energy, while the operation of the diesel engines remained stable, diminishing the ramp up and the steep gradients before the night hours' peak demand. Additionally, considering the system's ability for energy storage, as a result of the BESS installation, a portion of the PV energy produced in daylight time period could be shifted for later use and therefore the diesel engines could avoid abrupt load changes. Through the forecasting, it was possible to estimate an hourly based trajectory for the thermal energy generator operation and acquire the BESS setpoints which would result in the desired peak shaving and smoothing level. This analysis is followed by a simulation of the model to show how it works.

Aknowlegments

With the accomplishment of the present Diploma Thesis, I would like to thank my supervisor Professor Dr. Ioannis Antoniadis from the School of Mechanical Engineering of National Technical University of Athens (*NTUA*), Director of Mechanical Design and Automatic Control section, for his consideration while showing confidence to my skills. I would also like to thank Dr. Christos Yiakopoulos, member of the Dynamics and Structures Laboratory of Machine Design and Control Systems Section of NTUA, for all his assistance, encouragement and the guidance he provided me with during the realization of the present thesis. I also thank the people who supported me throughout my studies. Finally, I cordially thank my family for the undivided patience and support all these years!

1. Introduction

1.1 Background and motivation

More and more models are being built and developed nowadays around energy grid management and performance. These models aim to reduce the production of carbon dioxide and other chemicals that are harmful to the environment. Of course, an additional goal is to reduce the total cost of energy production but also as much as possible the contribution of renewable energy sources to electricity production. Renewable energy will play an even greater role as an energy source in the future. This will result in a reduction in the burning of fossil fuels for electricity and heat production with the associated CO2 reduction. Global warming and the dependence on fossil fuels primary energy sources will also be reduced. Energy production rates from renewable sources such as wind or sun is not in line with the rate of consumption. One, and probably the most important, handicap of most of these generation technologies is the non-manageability of their generation since they are solely responsible for production or climatic conditions, especially for wind and solar energy. The demand for management systems for the maximum exploitation of renewable energy sources is due to the fact that in 2050 the production of energy from renewable energy sources will exceed 60% of the production rate [1].The solutions, so far, seem to be focused on efficient battery energy storage systems (BESS). BESS can work in two ways. In one case it can work as a storage of excess energy, to secure it when there is a requirement for it. In the other case it can be used as a mechanism for smoothing the power generation curve. After a resources analysis and total consumption and production forecasting; a smart injection system has to be developed that control the energy flows. The injection system stores the energy in the BESS whenever there is a surplus in the generation from the PV system and uses it as an extra source of power when needed, for example during night. In case of Wind Energy, via an optimization algorithm the optimal curve is located and the BESS acts as a stabilizer of production on this curve. One challenge of finding a solution lies in the lack of generation data of big PV and Wind plants which are able to create perturbations in smaller grids. This data is needed to predict possible future behaviors of the plant and control the system voltage and power (both active and reactive) levels to avoid the grid perturbations.

1.2 Smart Grid Overview

Smart Grid is a concept for transforming the electric power grid by using advanced automatic control and communications techniques and other forms of information technology. It integrates innovative tools and technologies from generation, transmission and distribution all the way to consumer appliances and equipment. This concept integrates energy infrastructure, processes, devices, information and markets into a coordinated and collaborative process that allows energy to be generated, distributed and consumed more effectively and efficiently [2].

At the following figure, the basic topology of a SG with all the communication and data exchange pathways is presented. Common issues that are related with smart grids secure control and operation are related with grid stability (frequency-voltage regulation), balancing the power generation with the demand, the impulse renewable power penetration which needs to be directly compensated and the ability for self-sufficiency. These issues are commonly faced by load forecasting, predictive and optimization algorithms, while during the recent years, more and more energy storage systems are integrated into large smart grids.

Regarding the smart grids concept, a smart control and management system is necessary in order to achieve the most efficient and optimized BESS operation. Thanks to recent development in time-series forecasting and the possibility to access a big amount of data related with the power system operation, an EMS could implement a predictive strategy based on consumption/production forecasts and an objective function minimization. Therefore, load forecasting is a necessary stepping stone in order to achieve better energy dispatch planning, which is of great significance for the stable operation of conventional generators.

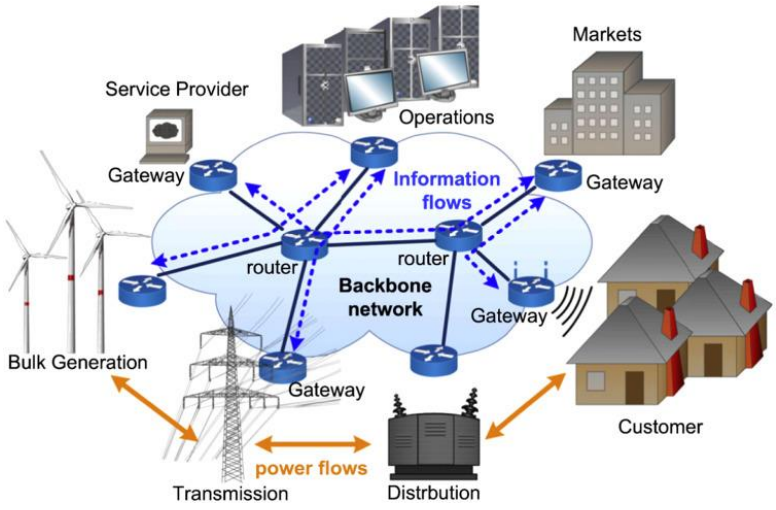


Figure 1.1 Smart Grid Topology

1.3 Aim of this study

Aim of this study is to build forecast models for the renewable energy sources and to develop a smart predictive control architecture for a BESS, based on load forecasting, for the optimum integration of PV and Wind power into an islanded power grid, through peak demand shaving and smoothening of the load curve and using a BESS system that will act as a stabilizer to smooth the Wind load curve. The forecasted parameters were the total load demand and the productions of thermal, solar and wind energy. The forecast was done with novelty neural network algorithms combining Convolutional and Recurrent Neural Networks. Compared to other studies, the present study differs as it studies the forecast of all energy sources, the management of this forecasted values through an intelligent algorithm that uses machine learning and then tests its operation in a model. The EMS algorithm, in concerning thermal and the contribution of solar energy, it is a predictive algorithm. It uses the forecast of the solar energy to define the degree of peak shaving the day before.

2. Backround

2.1 Machine Learning

Mitchell [4] define machine learning with this definition:

*A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P,if its performance at tasks in T, as measured by P,improves the experience E.*

This is a little bit confusing definition, but after we introduce some examples for task T, performance measure P and experience E, it will be make more sense.

The *task T* describes the task which should be fulfilled with the computer program. This could be a classification, regression, clustering or some more complicated task, like driving a vehicle. In this thesis the task T is a classification problem for both tasks. We have the task to classify the pose into eight directions and to classify the vehicle class.

The *performance measure P* is a measurement of the quality of the learned task. The performance measure P is depending on the task T. For classification it could be used the accuracy, which describes the ratio of correct classified examples with respect to all examples. In regression could it be the squared distance to the correct value, and in the task of driving a vehicle, it could be the average traveled distance until occurred the first failure. Sometimes could be used a measurement which penalize some mistakes not so hard like others.

With this measurement, we could evaluate the learned program or the model. Usually, we are inter- ested in the performance of the model on unseen data. But therefore, the model must be evaluated on an independent *test set*. This test set is separated from the training data and is never used for the training. Then we could do an estimation of the performance on unseen data.

The *experience E* describes source of the data which is used to learn. In a classification and regression task, it is a data set with input data and the desired output label to the input label. In case of the driving a vehicle it are a recorded sequence of images and steering commands. In this thesis, is the experience a huge amount of images, which shows one vehicle, and to every image exists two labels, which describe the pose and the vehicle class of the vehicle in the image.

A practical view of a machine learning system is depicted in Figure 2.1. The process is split into two phases. In the first phase, the machine learning algorithm is used to learn from the training data, and the second phase is the prediction. The training data could be labeled images of vehicles with the task to predict the pose. The machine learning algorithm learns

a model on these data and this model can then used to predict unseen images of the same task. This prediction is happen in the second phase and apply only the learned model. In our example, the learned model gets images of a vehicle and must predict the pose.

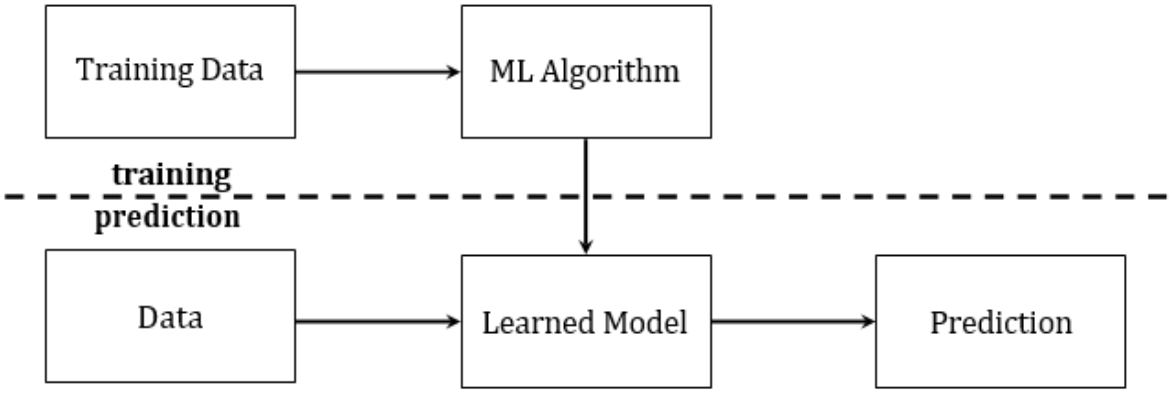


Figure 2.1 Workflow of a machine learning problem. The data will be used to train a model with a machine learning algorithm. Then, in the prediction phase, the learned model can be used to generate a prediction.

2.1.1 Learning Problems

2.1.1.1 Supervised Learning

Supervised learning methods [5] use labelled data and the goal of supervised algorithms is to determine a good approximation between a set of inputs and correct outputs. More formally, supervised methods approximate some function f that maps from some input space X to an output space Y, formally $f : X \rightarrow Y$. The iterative process of optimising this function to achieve a good approximation is known as the learning process.

An example is if we want an algorithm that classifies a specific object such as a ball. The input image has a corresponding output target that tells if the input image has a ball or not. By using labelled data, such an algorithm can iteratively be optimised to learn particular features describing a ball. Hence, for unseen images, the algorithm can correctly determine whether or not an image contains a ball.

2.1.1.2 Unsupervised Learning

Unsupervised learning methods [5] does not use labelled data, and there is no explicit description of how an algorithm should optimise the understanding of observations. Unsupervised methods are concerned with finding patterns or groups in data, for instance, by clustering similar observations which seemingly belong in the same categories. For

example, an unsupervised algorithm cannot tell if it is a ball or not. It may, however, group all similar images that potentially contains a ball in the same category.

2.1.1.3 Reinforcement Learning

Reinforcement Learning methods interact with an environment and must reach a certain goal. This could be to learn playing a game, or to driving a vehicle in a simulation. The algorithm gets only information how good or bad he interacts with the environment. For example, in learning to play a game, could this information the winning or losing of a game.

2.1.2 Under- and Overfitting

A machine learning algorithm must perform well on unseen data. The ability to perform well on unseen data is called *generalization*. The generalization error is measured on a test set. If a model perform not well on unseen data, then there are two reasons. Either the model has not enough capacity and *underfit* the underlying function, or it has too much capacity and *overfit* the underlying function. If a model is underfitting, then will be the training error and also the test error high. If the model is overfitting, then is the training error low and the test error high. The goal is to find a model, which has a low generalization error. The typical curves for training and generalization error are depicted in Figure 2.2.

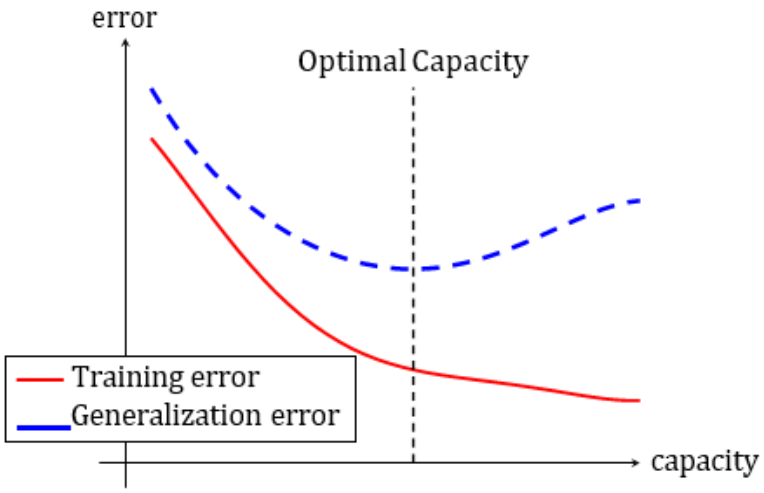


Figure 2.2 Shows typical curves for training and generalization error in dependency of the capacity of the model. Optimal capacity is reached at the minimal generalization error. Left of the optimal capacity is the model underfitting. On the right side of the optimal capacity is the model underfitting. On the right side of the optimal capacity is the model overfitting. The generalization error has typical a U-shaped curve.

In Figure 2.3 are three diagrams depicted, which shows the same noisy sampling of a sinus function. The samples are used learn a model, which describes the underlying function. The

left diagram learned a model with a low capacity. So, the training error is high and a test set would be also produce a high test error. In the center is depicted a model with a higher capacity. This shows a good approximation of the underlying function. Nevertheless, the model has a training error, because we sampled the sinus with noise. But this model will have on a test set the best generalization error compared to the other two models. The third diagram shows a model with a high capacity. The training error will be low, but the learned model is not a good approximation of the sinus function, which would result on a test set with a high error.

It is also important to use the right capacity of a model for the problem. And if we add right regularization, then it is possible, that we can use a higher capacity [6]. Because the regularization damping the capacity of the model.

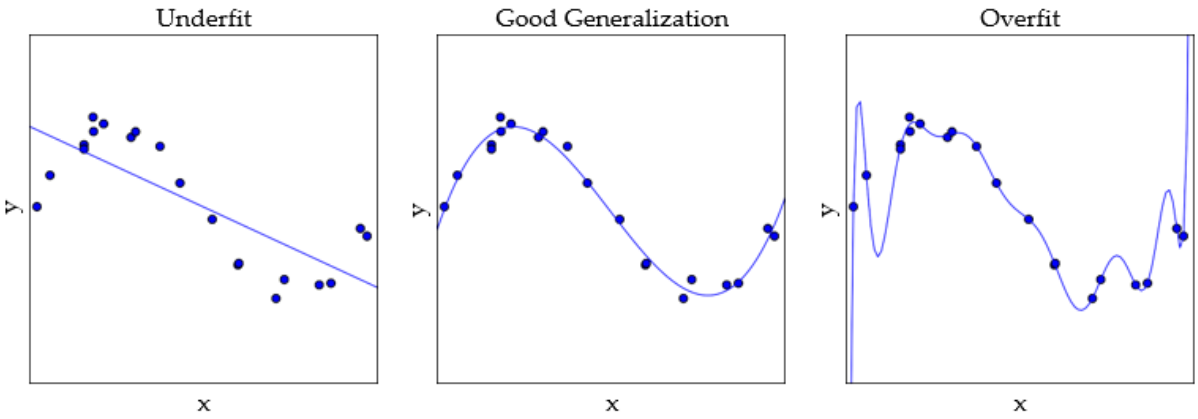


Figure 2.3 Theses three diagrams show three different models, which tries to fit the sampled points. The sampled points are sampled from a noisy sinus function. The models are described by a polynom of degree {1,4,15} . The left model underfits the underlying function. The right model overfits the underlying function. It has the smallest error to fit the sampled points, but on unseen samples, it will provide a bad error. Idea for this figure is from [6].

2.1.3 Artificial Neural Network

Artificial Neural Networks (ANN) are inspired by neuroscience. It is the attempt of mimicking the biological neural network (BNN) of a brain, to solve problems in the same way. The research of ANNs starts in the 1940s. Over the time, the ANNs had different periods of popularity and was further development. The last period is today, under the name of Deep Learning. After the winning of different competitions with ANNs, the popularity of ANNs are increased in the research community. The main characteristic of deep learning are the larger networks – deeper and wider – which are used to solve a problem. Goodfellow et al. [6] see two reasons for the success of deep learning:

1. More computationally power
2. Larger datasets

Today, the computers have more computationally power to train a bigger network in shorter time. So, it is possible to test different structures and different hyperparameters in shorter time. Furthermore, the larger datasets help to find a better generalization.

2.1.4 Feedforward Neural Networks

The feedforward neural networks or known under the name multilayer perceptron, are multilayered networks of perceptrons. Several perceptrons are connected in series. **Figure 2.4** depicts an example of a Feedforward Neural Network. A feedforward neural network consists of multiple *layers*, which have specific names. The first layer is called the *input layer*. The last layer is called the *output layer*. The other layers are the *hidden layers*. The hidden and output layer consists of several perceptrons, which are called *units.* The input layers contain only the input values. The *depth* of a network describes the number of layers, and the *width* of a layer describes the number of units. The depicted example network in **Figure 2.4** has the depth of two. The width of the hidden layers is four and the width of the input and output layers is two. The bias units are depicted with the +1, and will be not count to the width of the layer between every layer, the units are *fully connected* (FC). This means that every unit of layer has the output of all units from layer as input.

Through this concatenation of several layers, the feedforward neural networks gets the capability to solve non-linear problems, if the activation function is not linear. Would be the activation function a linear function, then has the network only the capability to solve linear problems, because the composition of linear function produce again a linear function. So the activation function should be a non-linear function.

Furthermore, a feedforward neural network has a very powerful property. With only one hidden layer with sufficient units and the right activation function, like the sigmoid function, the network can approximate arbitrarily closely every continuous function on a closed and

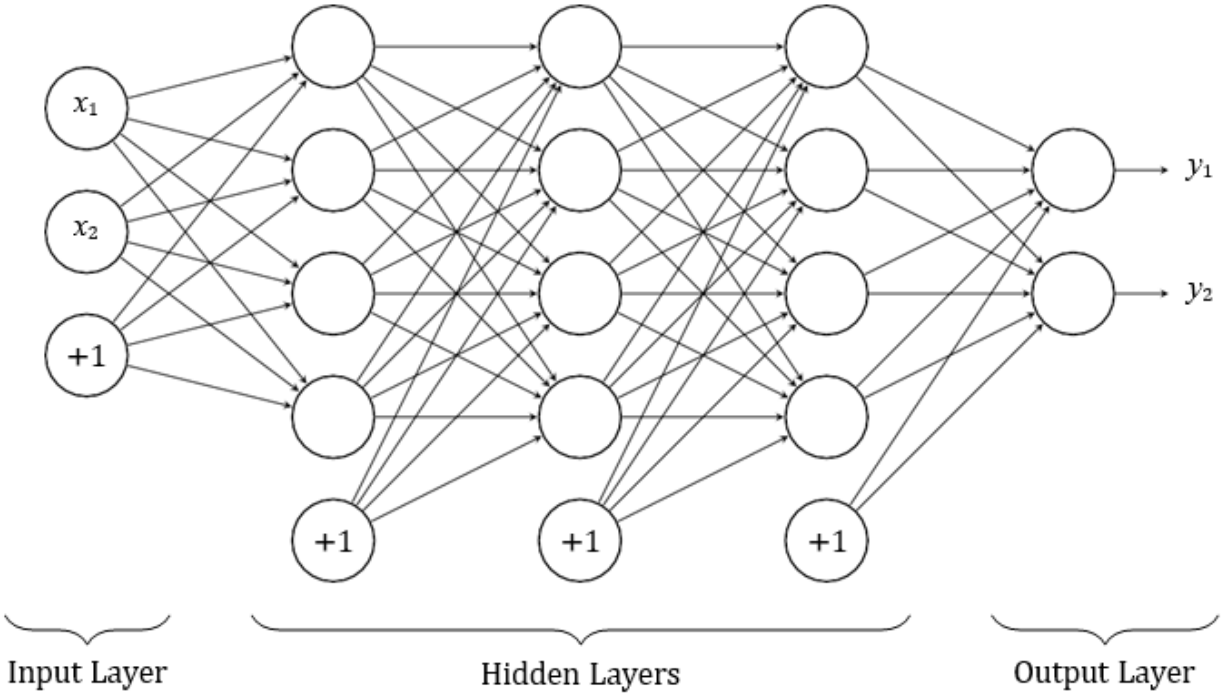


Figure 2.4 Feedforward Neural Network

2.1.5 Simple Neuron and Activation function

The activation function $\sigma(z)$ is the main part of a neural network, which gives the network the power to solve non-linear problems. If the units have no activation function or only a linear activation function, then the neural network could only solve linear problems. No matter how deep the neural network is. So the activation function should have a non-linear characteristic. A further property for the activation function is, that it must be continously differentiable. Otherwise, the backpropagation algorithm will not work, because it cannot compute the gradient.

In the beginnings of the research of neural networks, there were used saturated activation functions. This means that the activation was limited to (0,1) or (-1,1). As activation functions were used the *sigmoid* or *tanh* function. Both have the problem, that by small or big values of z the value of the gradient goes to zero, and the convergence of the learning decreases. In other words, the training of the neural network needs much more time. Another problem is the vanishing or exploding of the gradient, in case the network is depth. The vanishing

gradient occurs in earlier layers, because through the backpropagation will often be multiplied with small values. Many multiplications with small values between 0 and 1 tends toward zero. The exploding gradient happens if the weights are big, and the activation is near 0. At this point the derivative of sigmoid and tanh has reached their maximum.

The commonly known activation functions used in neural network models are: *Sigmoid, Tanh, ReLU, Threshold, Softmax*(**Figure 2.5**). The activation function, which have been used for this thesis is *ReLU.*

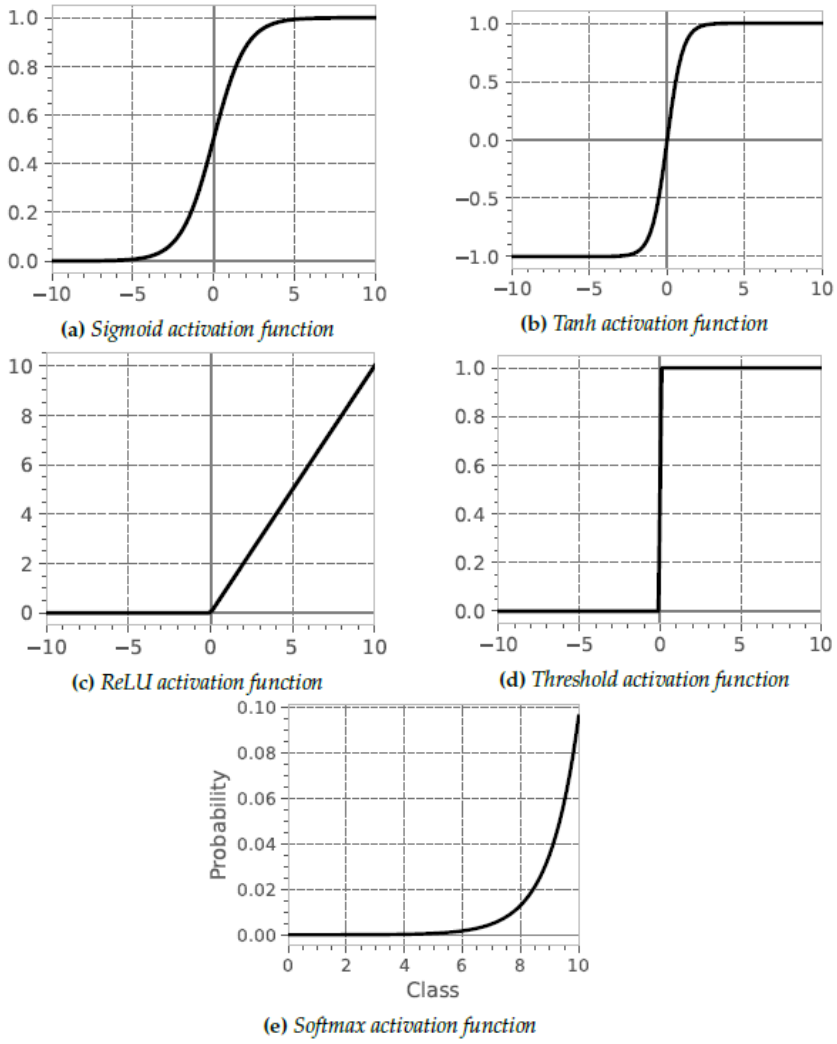


Figure 2.5 Commonly known activation functions used in neural network models

.

2.1.5.1 *ReLU* - Rectifiere Linear Unit

The *Rectifiere Linear Unit* (*ReLU*) is the most used activation function for neural networks. The formula for this function is simple

$$\sigma(z) = \max(0, z) = \begin{cases} 0, & if\ z < 0 \\ z, & if\ z \geq 0 \end{cases}$$

This means, that the function is 0, if *z* is negative otherwise the value is *z*. The *ReLU* activation is in **Figure 2.5** depicted. The interesting fact is, that the function is for positive values linear and not saturated. A further property, which makes *ReLU* so popular is the fast computation of the derivative. The derivative is either 0 (at negative values) or 1 (at positive values). Krizhevsky et al. [7] have applied *ReLU* in combination with Convolutional Networks and have a six times faster convergence as with the same network with the activation function tanh. Batch normalization with *ReLU* is heavily used in modern network architectures [8,9]. We introduce batch normalization in the next section, but *ReLU* with batch normalization increases the learning speed of the network. This comes from the normalization of the batch normalization, so that the mean is zero. The input with zero mean increases the convergence [9].

ReLU has a problem with dying units. This is the case, if no example in the training set can activate the unit. If this happens, then is the unit dead and cannot more activated, because the gradient is forever 0 and the weights are not more changeable for this unit. To counteract the dying units, it is possible to use leaky *ReLU*, which change the behavior for negative activation. It adds for the negative z a linear component, with the slope $0 \leq \alpha \leq 1$. So, the gradient is never zero.

2.1.6 Regularization

Regularization are methods to control the overfitting or rather to improve the generalization error. There are different methods, how to apply regularization. Some of the methods are common approaches in machine learning, and other are only usable for neural networks. Goodfellow et al. [6] argument, that large models, that has been appropriately regularized, find the best fitting model. At this point, we show three regularization methods, which are used in this thesis.

2.1.6.1 Early Stopping

On training of large models, normally the training and validation error decreases over the time, but at one point the validation error starts to increase. At this point the model is starting to overfit and learn specific properties of the training set. To stop at this point, it is

applied the method of *Early stopping*. It returns the model, which has the lowest validation error. Therefore, the training needs a validation set, to evaluate periodically the validation error. The normal period is after every epoch.

But how could we ensure, that this increasing was not caused by noise? The training is not stopped after the first increasing of the validation error. The network is further trained until a threshold of "number of epochs without improvements" is reached. Through the evaluation of further epochs, we get the trend of the validation error for more training. For example, if 10 times in a row, the validation error has no improvements compared to the best validation error, then is the training stopped, and the model with the best validation error is returned.

2.1.6.2 Batch Normalization

The training of a network changes the weights on every layer. This change has the effect, that the input distribution changes during updates of previously layers. The authors of *batch normalization* [20] called this effect *internal covariate shift*. To counteract the change of the distribution during the learning, they introduced the batch normalization. Every minibatch, which is inserted in the network, will be on every layer normalized on the input of the previously layer. The formula for the normalization is the following

$$\mu \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\sigma^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu)^2$$

$$\tilde{x}_i \leftarrow \frac{x_i - \mu}{\sqrt{\sigma^2 + \varepsilon}}$$

where μ and σ describe the mean and the standard deviation, xi is the normalized value of the input. The value m describe the size of the minibatch. The value x has then a mean of 0 and a standard deviation of 1. The batch normalization will be applied after the linear combination of a unit.

To not lost the representation power of the units, the value xi will be scaled and shifted

$$y_i \leftarrow \delta\tilde{x}_i + \beta$$

where δ and β are learned parameter during the training. So, the value yi could have any mean and standard deviation. This is useful in the case of using sigmoid as the activation

function. Without the scaling and shifting, the activation function will be act as an almost linear function, because through the normalization, the input is scaled to a range, on which the sigmoid function is almost linear. At test time, μ and σ are replaced with the average, which is collected during the training. So, it is possible to predict a single example, without to have a minibatch. Through the applying of batch normalization, the learning rate can be increased, and this results in a faster training. Furthermore, the accuracy is increasing compared to the same network without batch normalization [9] The batch normalization helps the activation function ReLU to learn faster and have a better performance.

2.1.6.3 Dropout

Dropout was developed by Srivastava et al. [10] and is a regularization method for neural networks. The key idea is, that units will be randomly dropped for every iteration of the training. This means, that the dropped units and their connection are zero. This should prevent the co-adapting of the units. During the testing the dropout is not available, and all units are used for the predicting. Srivastava et al. [10] tested dropout on different datasets and they had on all an improvement of performance. The drawback of dropout is the increasing training time, because not all weights a trained in one iteration. If one unit is dropped, then are all depending on gradients are zero and therefore the corresponding weight will be not updated. Dropout could be seen as a training of a subset of the model. Every iteration builds a different version of the model, and every weight is updated with another set of weights. This prevents the co-adapting of the units, because a unit cannot more rely on that another unit is available. A similar dropout is the Spatial Dropout, which is introduced by Tompson et al. [11]. It is used in convolution neural networks to drop complete feature maps and not only single units. This brings us to the next section, where we introduce convolution neural networks.

2.1.7 Recurrent Neural Networks

Consider a simple neural network, but where nodes in each layer now have in-between connections (recurrent connections). The result is a recurrent neural network (RNN) as shown in **Figure 2.6**, where the connections represent temporal dependencies (dependencies in time) and introduce an extra set of optimizable weights. RNNs are another family of neural networks designed for applications on sequential problems like language processing and the most simple RNNs only introduces these recurrent connections.

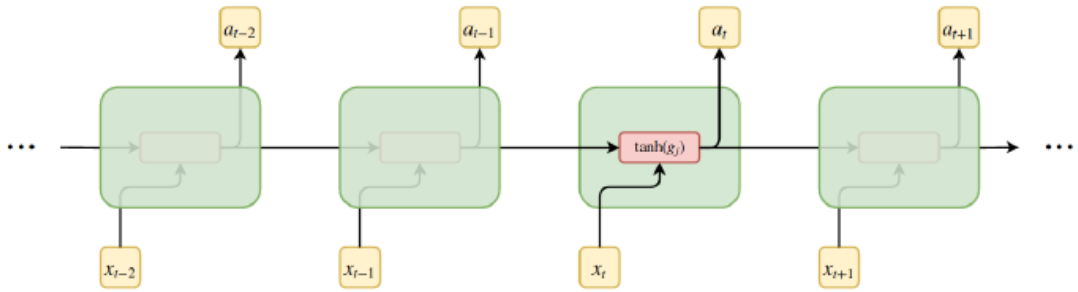


Figure 2.6 An illustration of 4 standard recurrent units in a layer with their respective recurrent connections. The output signal is a function of the current input and recurrent input at time t and t-1, respectively.

Formally, as shown in **Equation below**, we let the input at time *t* be defined as x_t and recurrent inputs from the previous timestep be defined as a_{t-1}, both being vectorized inputs. Similar to the weighted sum we described earlier, the latter denotes the signals of the weighted sum g_{t-1} of the previous recurrent unit. By introducing recurrent inputs, we introduce a new set of weights. We let the input weight set be defined as b_{ih} and the recurrent input weight set be defined as b_{hh}, respectively denoting input hidden and hidden-hidden weights.

$$g_t = \beta_0 + [\beta_{ih} * x_t] + [\beta_{hh} * \alpha_{t-1}]$$

$$\alpha_{t-1} = tanh(g_{t-1})$$

2.1.7.1 The learning process

The learning process in recurrent networks is similar to earlier definitions for feedforward networks. Signals propagate forward through the network and tanh is usually applied for non-linear activation of the weighted sum, hence $\alpha_t = tanh(g_t)$ being its specific form. Lastly, because RNN units are recurrent applications of themselves, the backpropagation of gradients is an ordered operation through each timestep. This is an extension of the ordinary backpropagation algorithm known as backpropagation-through-time (BPTT)

2.1.7.2 Long Short-Term Memory units (LSTM)

Long Short-Term Memory (LSTM)-based networks use carefully designed LSTM cells. These are computational units in networks controlling information flow through gating mechanisms. Initially proposed by Hochreiter and Schmidhuber [12], the architecture was designed to overcome the above-mentioned problem of learning long-term dependencies. One LSTM unit is composed of multiple gating mechanisms, namely, the forget gate, output gate and update gate. Collectively, they enable a possibility to maintain the overall cell state in each unit such that a LSTM layer is able to effectively capture longer temporal dependencies. As the names imply, the subset of information to be removed from the cell

state is determined by the forget gate and the new information to add to the cell state is controlled by the update gate. The output gate is concerned with controlling the output flow from the cell state, and thus determines what information is passed onwards to the succeeding LSTM-unit.

2.1.8 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a family of neural networks that are commonly applicable to problem domains where spatial information and other grid-like topologies are of importance. The most common application is image classification, as images can be considered 2D-grids. However, CNNs are also applicable on time-series [13], which in essence are 1D grids with samples at fixed time intervals. Overall, the main difference between standard feedforward networks and CNNs, is the architecture. Consider a standard feedforward neural network about neural networks. There are two concerns when applying a feedforward network to grid-like topologies with more than one dimension. The first is that the spatial information is not preserved, as the grid has to be flattened to the networks. Secondly, an application like this does not scale for use cases like image classification and other grid-like topologies, due to the increased number of parameters as a result of many interconnected neurons on the grid. An increase like this would require more computational resources, as the optimization problem require more effort. CNNs solves these problems by only looking at a subset of the input, known as local connectivity. The number of weights are reduced and spatial information is still preserved by applying two additional mechanisms; the convolution- and pooling operation.

2.1.8.1 The learning process

Inputs propagates forward in the network and generates some predictions. Backpropagation and forward propagation of signals is conceptually equal to vanilla neural networks. However, the difference in CNNs exists in their architecture, where convolutional- and pooling layers are introduced. The general scheme of a CNN is shown in **Figure 2.7.**

2.1.8.2 Convolutional Layers

Convolutional layers in a CNN takes advantage of the convolution operation, a mathematical operation on two functions, which generates a third function explaining an estimated relationship between both [6]. In the context of grid-like topologies, the convolution operation can be considered as a sliding window over the grid input, usually referred to as convolving a filter or kernel on the input. **Figure 2.7** illustrates the convolution operation in CNNs. Each convolutional layer contains a set of learnable filters, more specifically a set of weight matrices. The number of learnable filters is equivalent to the number of nodes in a layer, as each node represents a filter learning different features. The filters are spatially small and are convolved on the input by computing a weighted sum to generate an output volume, the set of feature maps. Because one feature map is a linear activation, each map

is passed through a non-linear activation function, similar to the process in feedforward vanilla networks. However, the output

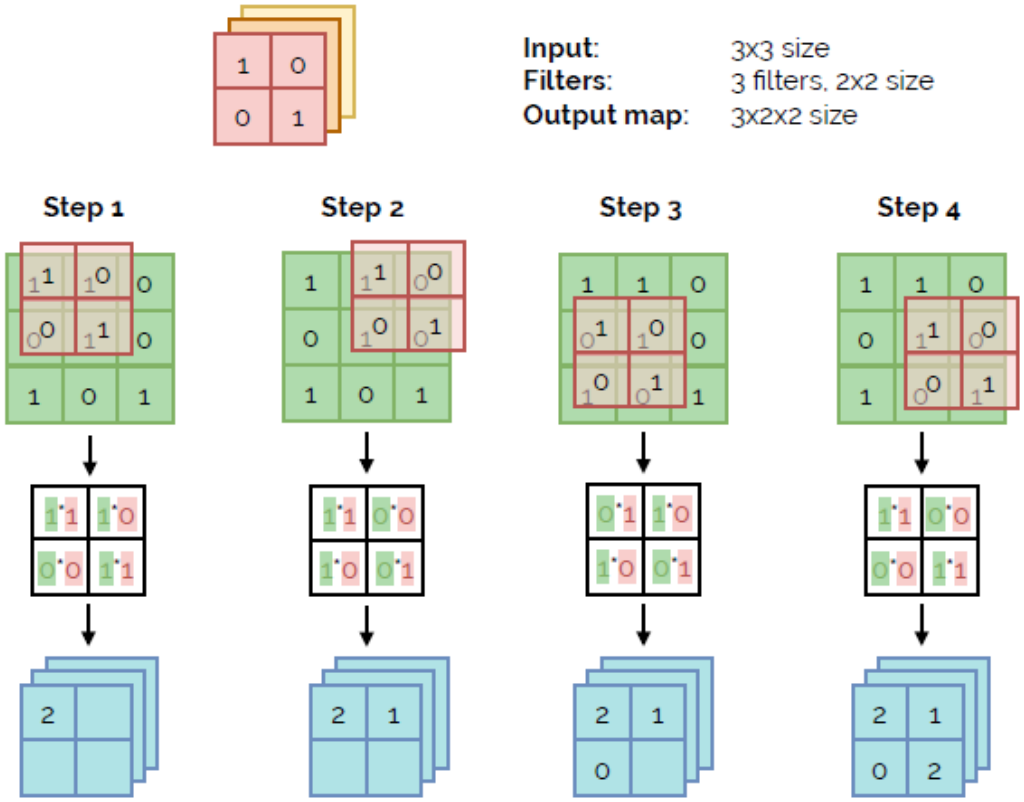


Figure 2.7 Illustration of the convolution operation in CNNs for 2D-grids. The operation is performed for each filter on the input, resulting in an output volume with the same spatial size as the number of filters and the filter size. Three 2x2-filters, thus results in an output dimension of 3x2x2.

volume of a convolution layer is not fixed. The process of adjusting the output volume is dependent on multiple factors, introducing some additional hyperparameters. For instance, step size (stride), filter bank (number of filters), filter size, padding and dilation rate are additional mechanisms in the convolution layer that determine depth and spatial dimensions of the output volume. We will not describe these in detail in this thesis, but their importance in the process of model optimization should not be disgarded. Tuning these hyperparameters share one common motivation, which is to reduce the model complexity by reducing the number parameters and increase local connectivity. This enables filters to look at larger proportions of input, without introducing additional computational complexity.

2.2 Related Work

This chapter presents the related work and theoretical background. In the first section, it is presented related work of comparative studies of RNNs and CNNs. Further on, it is presented the general concept of forecasting and a comparison between the most used method for forecasting, ARIMA, and Deep Learning.

2.2.1 An Evaluation of Convolutional and Recurrent Neural Networks

This study presents the most extensive comparison of RNNs and CNNs. In their paper, Bai et al. [14] presents an empirical comparison of different RNN architectures against CNN on various sequence modelling tasks. They address the question of whether the recent successes of CNNs on sequential problems are specific to the studied application domains, or if they are applicable to sequence tasks in general. In their paper, they study the effect of CNNs on various problems in which they conclude how CNNs are indeed applicable to different domains. Their CNN, named TCN for Temporal Convolutional Network, outperforms canonical recurrent architectures like LSTM and GRU. Through the experiments, they use the same configurations with various kernel sizes and layers. Dilated convolutions are used with Adam optimiser and a learning rate of 0.002. For the RNNs, automatic hyperparameter optimisation is applied, where they use grid search to find optimal configurations. Moreover, the study further analyses the effect on how the TCN captures temporal patterns and how the memory mechanism works. Overall, they find that the CNN-based network shows the ability to capture long history more efficiently than the RNNs as well. While this comparison study is essential to our research, the effect on time-series is not explored in their paper. This further motivates for our proposed research question, in which we extend the study to understand how CNNs can be used for time-series classification.

2.2.2 Time series forecasting: ARIMA vs. Deep Learning

The aim of time series analysis is to study the path observations of time series and build a model to describe the structure of data and then predict the future values of time series. Due to the importance of time series forecasting in many branches of applied sciences, it is essential to build an effective model with the aim of improving the forecasting accuracy. A variety of the time series forecasting models have been evolved in the literature. Time series forecasting is traditionally performed in econometric using ARIMA models which is generalized by Box and Jenkins [15]. ARIMA has been a standard method for time series forecasting for a long time. Even though ARIMA models are very prevalent in modeling time series they have some major limitations [16]. For instance, in a simple ARIMA model, it is hard to model the nonlinear relationships between variables. Furthermore, it is assumed that there is a constant standard deviation in errors in ARIMA model, which in practice it may not be satisfied. When an ARIMA model is integrated with a Generalized Auto-regressive Conditional Heteroskedasticity (GARCH) model, this assumption can be relaxed.

On the other hand, the optimization of an GARCH model and its parameters might be challenging and problematic [18]. ARIMA usually only deal with univariate data. In real world, it can be seen many datasets have multiple inputs. For instance, if the wanted is to predict the air pressure in one area, it is also helpful investigate the air pressures in other areas because they might have a potential effect on the given area. Finally, ARIMA models usually do not work well in long term forecast, like in this thesis. On the other hand, RNN and CNN models have multiple advantages. They have the ability to approximate arbitrary nonlinear functions. Furthermore, they can handle noise. Neural networks are super robust to noise and they can even learn with the existence of missing values. CNNs and RNNs accept multivariate inputs. Any number of features can be sufficed.

3. General informations

To evaluate CNN, LSTM and hybrid method and provide a comparison, we develop a system testbed to perform experiments. In this chapter there is a description of the design decisions in terms of model development and hyperparameter selection and the process from data analysis to experiment execution.

3.1 Dataset

The dataset used for training and testing of all models collected from three years in an island. The provided data concern electric production from three sources (Thermal, Wind and Solar Energy), the various climatic conditions that prevailed and the Total Energy Consumption. As previously assumed, the first task of this thesis is to forecast the total energy production and consumption. As first step, the data were separated and by observation of the interdependencies as well as the literature, the appropriate input values were selected for the each output value.

3.2 System Specifications

The Python programming language [19] is used to perform data analysis, develop models, run experiments and evaluations. With a rich ecosystem and a diverse set of supported libraries and frameworks, tasks related to data analysis and neural network modelling are more convenient. Further on, Table 2.1 summarises what software, hardware and frameworks that are used.

Software		
Name	**Version**	**Description**
MS Windows 10 Home	10.0.18363	Operating System
Python		Used for implementation
Keras		Used for nuilding models
Pandas		Used for data analysis
TensorFlow		Used as backend for Keras
Cuda		Required for Tensorflow
Hardware		
Name		**Description**
CPU		Intel i5-8300H
GPU		NVIDIA GTX 1050Ti
Memory		8.00GB
GPU Memory		4.00GB

Table 3.1 System specification of hardware and software

Pandas [20] have most direct applications in this thesis. The former is extensively used in the field of data analysis and especially for time-series analysis. Keras however, is a high-level

framework for developing neural network models. It is designed to be built on top of existing platforms like TensorFlow [21], which is used as the backend for Keras and facilitates for computational efficiency. In this thesis, the GPU-release of Keras is used, enabling GPU-optimised computations. The corresponding drivers and interfaces responsible for the underlying interaction between TensorFlow and hardware require additional installation and configuration. CUDA [22] is additional underlying requirement when installing Keras with TensorFlow.

3.2.1 TensorFlow

TensorFlow is an open-source framework designed for large scale numerical computations. It has particular support for machine learning and deep neural networks and is supported on most platforms. Additionally, there is an extensive open-source community actively engaging in the development of TensorFlow. In general, TensorFlow represents computations in a computational graph, where nodes represent operations and edges represents tensors transitioning from one state of the graph to another. A tensor is a multidimensional array flowing in-between each operation. This model of computing is referred to as the dataflow paradigm, where information is a functional transformation of operations.

In terms of architecture, TensorFlow is implemented as a layered architecture. Applications and libraries like Keras utilize underlying computations, hardware interactions and optimizations implemented in the TensorFlow core/kernel as an application on top. For instance, implementations in Python and C++ are initial implementations whereas other examples include JavaScript, Java and Go support.

3.2.2 Keras

Keras is a machine learning library primarily designed for neural network modelling. It is also open-source and has gained popularity in recent years. Keras has a simple and intuitive interface for the development of neural network models. Moreover, the library can be run on top of multiple machine learning platforms like TensorFlow and Theano and reduces the threshold of complexity when developing neural network models. Overall, the framework works well as a high-level application interface.

Keras has a modular implementation of core elements in neural network models, which includes different optimizers, layers and layer types, activation functions, metrics and regularizes.

In general, Keras functions as an abstraction layer on top of the more technical platforms like TensorFlow. The application interfaces are consistent and intuitive, which provide modular functionality. For instance, implementations of the more complex gating units in recurrent networks, like LSTM-units, are applied in this thesis. Such modules are also easily

extensible, facilitating for custom implementations as well. In this thesis, however, custom units and implementations were not required, and thus not used.

3.2.3 Pandas

Pandas is a library used for data analysis. It is open-source and provides a good interface for advanced analytics and data manipulations. The library is tailored well for time-series analysis in particular. For instance, Pandas has an interface for grouping and aggregating observations, upsampling/downsampling of a time-series and well-defined interfaces applicable to data interpolation and cleaning.

Moreover, Pandas takes advantage of structured data, tabular datasets and vectorized operations and represents the data in Pandas data frames or Pandas series. Each data type has its own set of data manipulation operations. For time-series data, Pandas is also able to automatically infer and parse timestamps and creating an index so that operations on a timeseries becomes a less tedious task. While Python provides similar parsing features, this functionality is highly preferable and less time consuming. Many of these features are used extensively in this thesis, especially for data analysis. They lower the complexity of exploratory data analysis, making it easier to derive descriptive statistics and intuitive visualisations. We aggregate, filter, clean and resample the time-series with Pandas, which would otherwise become a manual task. Additionally, Pandas is used for generating descriptive statistics and figures through the built-in plotting interface. The plotting interface for Pandas is built on top of Matplotlib, a Python library used for creating figures. Matplotlib is used in the thesis for generating figures.

3.3 Photovoltaic system and parameters

To understand how does a PV cell work and consequently to understand the operation of the PV panel, two concepts are important: solar radiation and irradiance. solar radiation is defined as a solar energy that reaches the earth in the form of electromagnetic radiation. this can be divided into two categories the direct radiation and diffuse radiation. be first directly he eats a horizontal surface on the planet through a straight path traveled by sunlight. The second corresponds to the light that hits the surface indirectly, that his, after being reflected and diffractive during its course.

The solar radiance or irradiation is a measure of solar radiation its units are W/m^2, that means that the irradiance measures the solar radiation power in relation to the area. The value of 1000 W/m^2 is used as a reference to evaluate PV panels performance.

Figure 3.1 show the basic model of a photovoltaic cell. This model was built based on those presenting by [23].

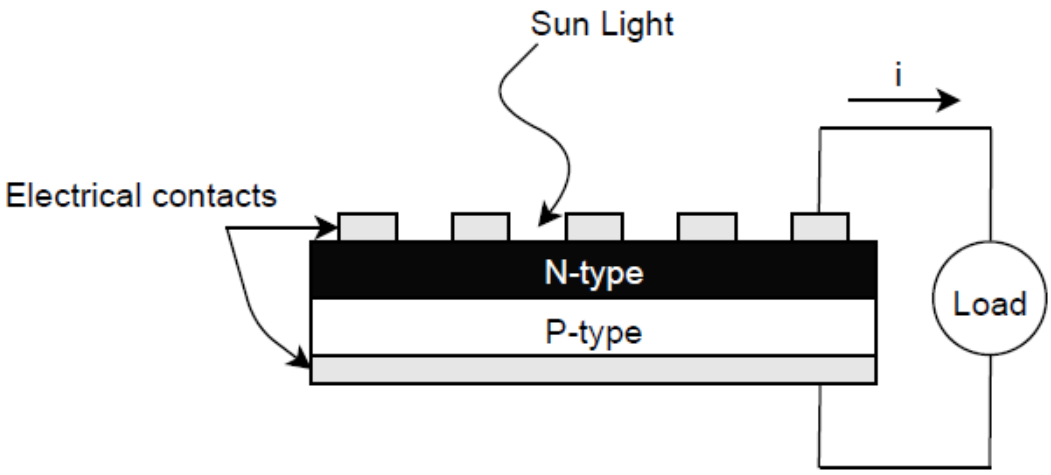


Figure 3.1 PV cell model

Analyzing the **Figure 3.1**, one can notice that the football to Excel is formed by the union of two types of semiconductors materials: the N-type and P-type. there are also electrical Contacts on both sides of the cell, used to closing the electrical circuit in which the current flows. The operation of the effort will Accel is based on for the voltaic principle. This principle describes the process of transforming the incident solar electromagnetic radiation on the surface of the cell into electricity.

Another climatic variable that influences affordable like generation is temperature. The change in temperature causes changes in the value of the open circuit voltage of a panel. The higher are the temperatures, the lower will be the voltage values, that is, the dependence is inversely proportional as shown in **Figure 3.2**. Given this, one can see how climatic variables influence the generation levels of photovoltaic system. Considering that the behavior of such variables is chaotic, it is inferred that the generation of photovoltaic energy also exhibits a volatile behavior. **Figure 3.3** shows the one- day generation profile of a real photovoltaic system installed in the examined island. It can be seen the influence of the intermittent behavior of the football taxers and how abrupt variations occur in the profile, due to changes in the climatic variables.

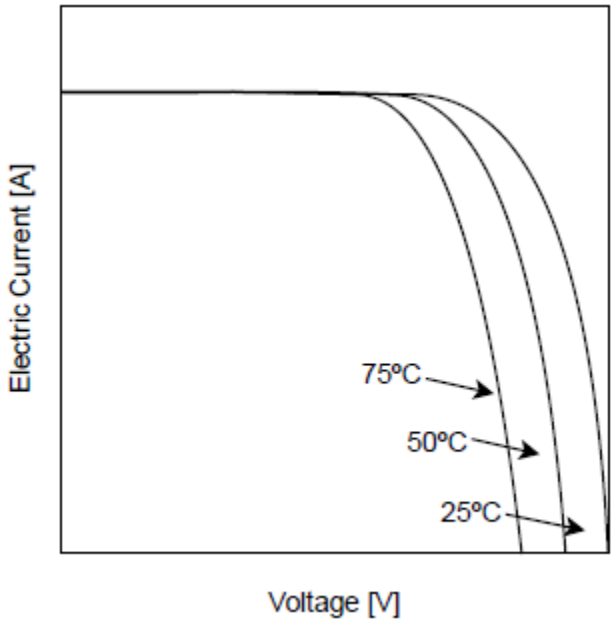


Figure 3.2 Temperature influence over the photovoltaic power generation [24]

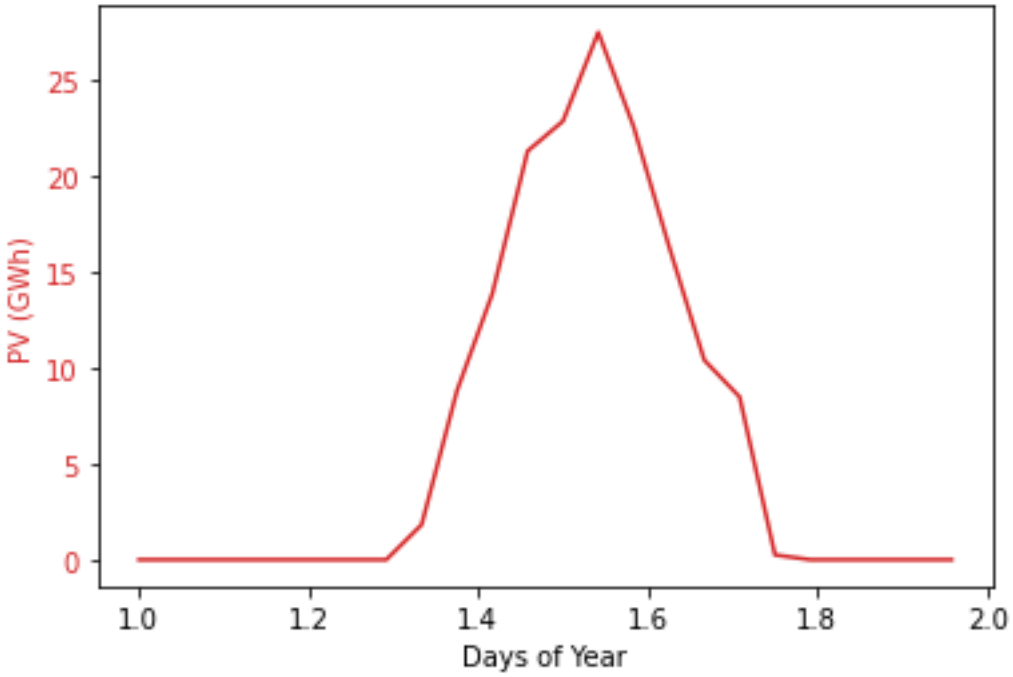


Figure 3.3 PV innovation daily profile of the PV system installed in the examined island

4. Forecasting

A forecasting model is a necessary subsystem that needs to be implemented in a predictive energy management algorithm capable of compensating future events. In this chapter the forecasting methods is going to be analyzed. It will be analyzed separately for each means of production as well as for consumption. It should be noted that since the final issue is energy management algorithm, which has to be managed of batteries and electric vehicles, the forecast is made in the depth of one year, also knowns as Long-Term, with the data of the previous two years as a training unit of neural networks [25]. At this point, it is crucial to clarify that the provided dataset, is real measurements of consumption and energy production of an island from 2014 to 2016. In the case of a larger database, perhaps even better results would have been achieved than what will be presented below. The first step for developing a neural network model concerns the data preprocessing. This refers to a set of actions that need to be applied to the dataset in order to divide the dataset to appropriate subsets and prepare the data format, in order to be inserted to the training algorithm in each case separately. In addition, below will be listed CNN, LSTM and Hybrid models(CNN and LSTM combination).

4.1 Input Values and Training Description

4.1.1 Total Load Forecast

The first thing that needs to be predicted is the total load required. This as shown in the **Figure 4.1** below differs both from season to season and there are strong variations during the day. The latter will concern in the next chapter, when the EMS algorithm will be analyzed. A strong correlation between the trend of the load curve and the temperature data of the island was observed, making the latter as an appropriate input for the developed neural network model. The hourly temperature data was produced by the long validated CFSR numerical weather model (NWM) from representative grid points near the most-highly inhabited areas of the island for the years examined, so that the correlation of weather phenomena with electricity consumption to be intensified. To find the best netwotk's structure all the possible models were built and tested. The possible structures are demonstrated in the **Figure 4.2**. The figures are representative and do not show the exact form of the models. The LSTM, CNN and Hybrid models were constructed from four, three and six hidden layers respectively. The inputs of the neural network were determined based on common input variables for similar networks referred in load forecasting studies and after a trial-and-error iterative procedure. The inputs are summarized to the following:

i. 48 values of the hourly consumption data of the two previous days,

ii. 24 values of previous day temperature data

iii. 7 binary values corresponding to the day of the week

iv. The network output consisted of a 24-variable vector containing the next day's forecasted load values.

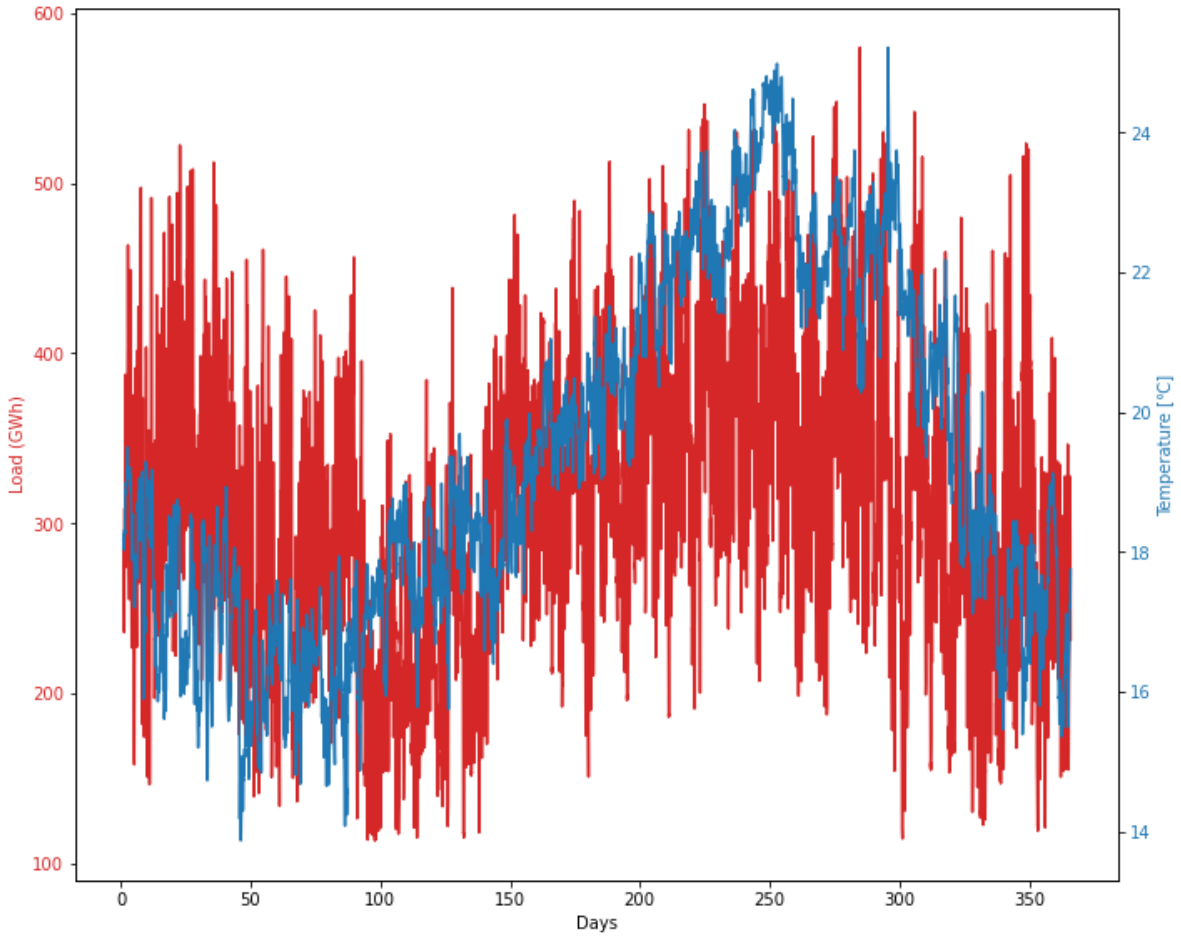
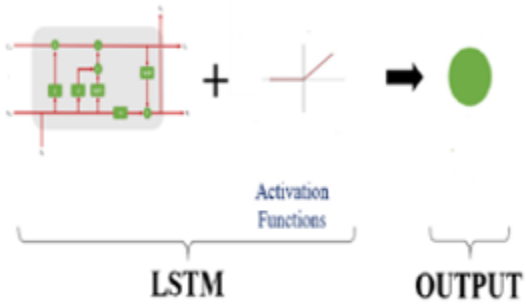
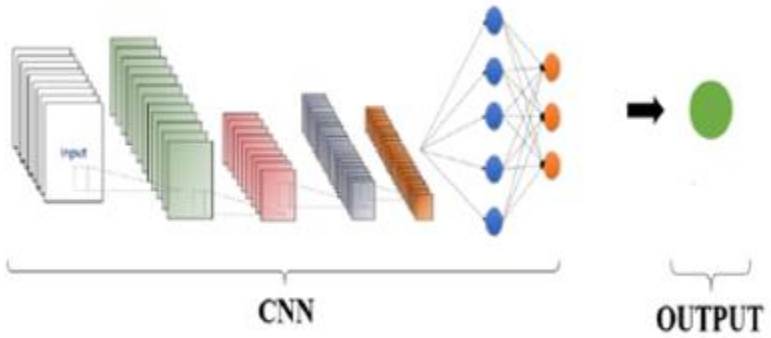


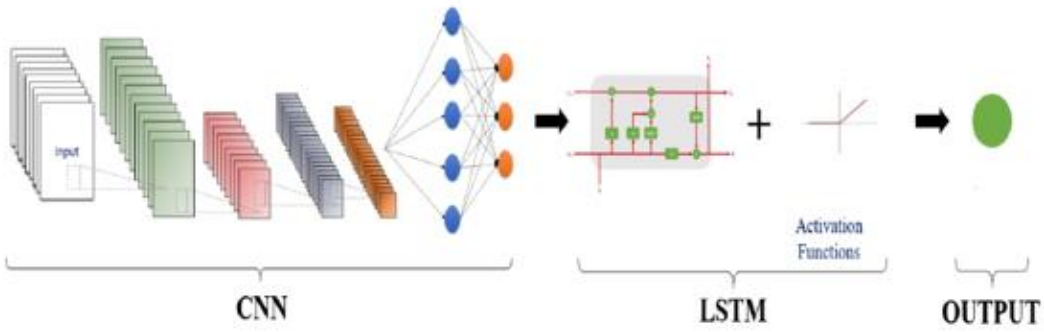
Figure 4.1 Yearly time period load curve correlation with temperature, for the test case system



LSTM Model



CNN Model



CNN-LSTM Hybrid Model

Figure 4.2 Neural Network models

A similar issue was studied by Chapaloglou et al. [2] so after checking the same input parameters were selected.

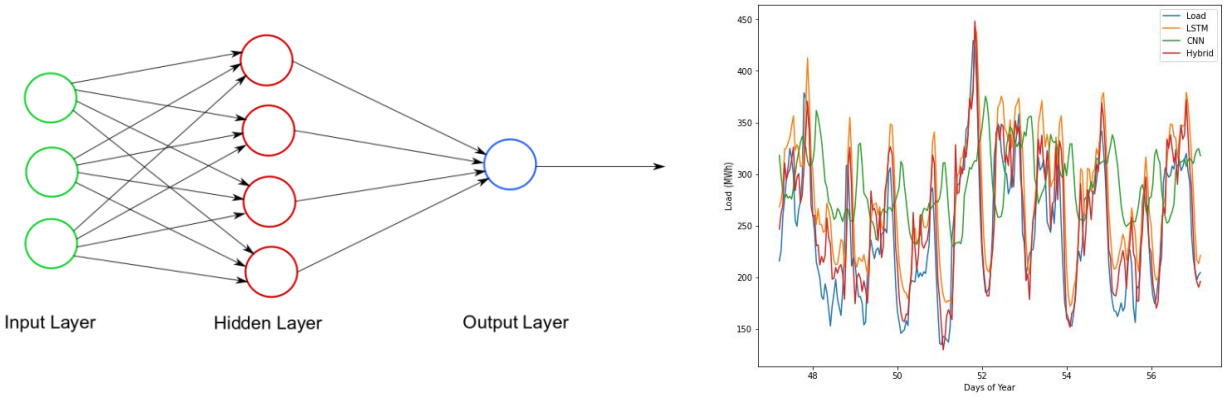


Figure 4.3 Developed network structure and data flow

Before the training precedure it has been used MinMax Scaler function to be scaled between [0,1].

$$MinMax\ Scaler = \frac{x_i - mean(x)}{\max(x) - \min(x)}$$

4.1.1.1 Network Training

As mentioned earlier, a training-test split is typically required for an ANN forecast. The test set has to follow two conditions: First of all, it must be large enough to yield statistically meaningful results. Secondly, the test set needs to be representative of the data set as a whole. In other words, this set of data must not contain different characteristics than the training set. Assuming that the test set meets the preceding two conditions, the goal is to create a model that generalizes well to new data. The first issue here is the division of the data into both sets. Although there is no general solution to this problem, several factors such as the problem characteristics, the type of data and the size of the data set should be considered while making the decision. The literature offers little guidance in selecting the training and test samples. In general, the sample size is closely related to the required accuracy of the problem. The larger the size, the more accurate the results will be. For this particular problem, the models and the selected network hyperparameters did not represent a high complexity, hence, the data size was not a limiting factor of the accuracy achieved. Because the dataset was pretty large, it was achievable to have a 70/30 split, 70% of the total dataset used for the training and 30% for the testing. As a result, the training procedure took place in the years 2014 and 2015.

For the purpose of developing the neural network applied in this study, a supervised learning method was implemented. With this methodology of training, the input stimulus that it is applied to the network's neurons results in an output response which is compared with a prior desired output, namely the target signal. If the actual response differs from the target response, the neural network generates an error signal which is then used to calculate the appropriate correction that should be made to the network's synaptic weights. This is

repeated until the actual output matches the target output, considering a validation subset of the dataset, whose output should be in an acceptable error range compared to the target response. The training method is similar in all neural networks that have been used in this study so there will not be additional analysis in the training.

4.1.2 Solar Energy Forecast

Solar energy is a popular renewable energy source because it is abundant and environment friendly. A challenging issue associated with the solar PV is that its power output strongly depends on uncertain and uncontrollable meteorological factors, such as atmospheric temperature, wind, pressure, and humidity [26]. As will be shown in the **Figures** below, PV energy is directly related to Direct Solar irradiance and temperature, as it will be seen there is a seasonal variation. Although, a numerical linear correlation will be presented below.

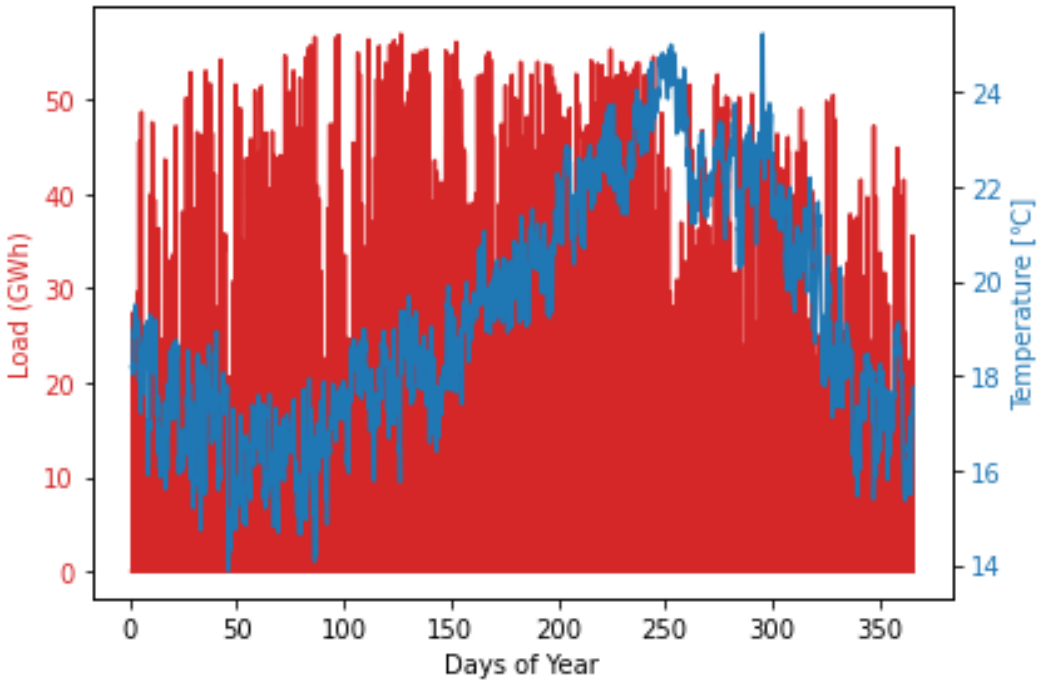


Figure 4.4Correlation between PV Energy and Temperature

The solar irradiance and PV power forecasting methods are divided into physical and statistical models. The physical model mathematically or numerically manages the interaction of solar radiation in the atmosphere based on the laws of physics. It comprises numerical weather prediction, sky imagery, and satellite image models. The statistical model finds a relationship between input and output variables and consists of conventional statistical models and machine learning models. Conventional statistical models include the fuzzy theory, Markov chain, autoregressive, and regression models. The machine learning model, also known as an artificial intelligence model, can efficiently extract high dimensional

complex nonlinear features and directly map input and output variables. In the past, the well-known machine learning models for predicting solar energy were the support vector machine (SVM), k-nearest neighbors, artificial neural network (ANN), naïve Bayes, and

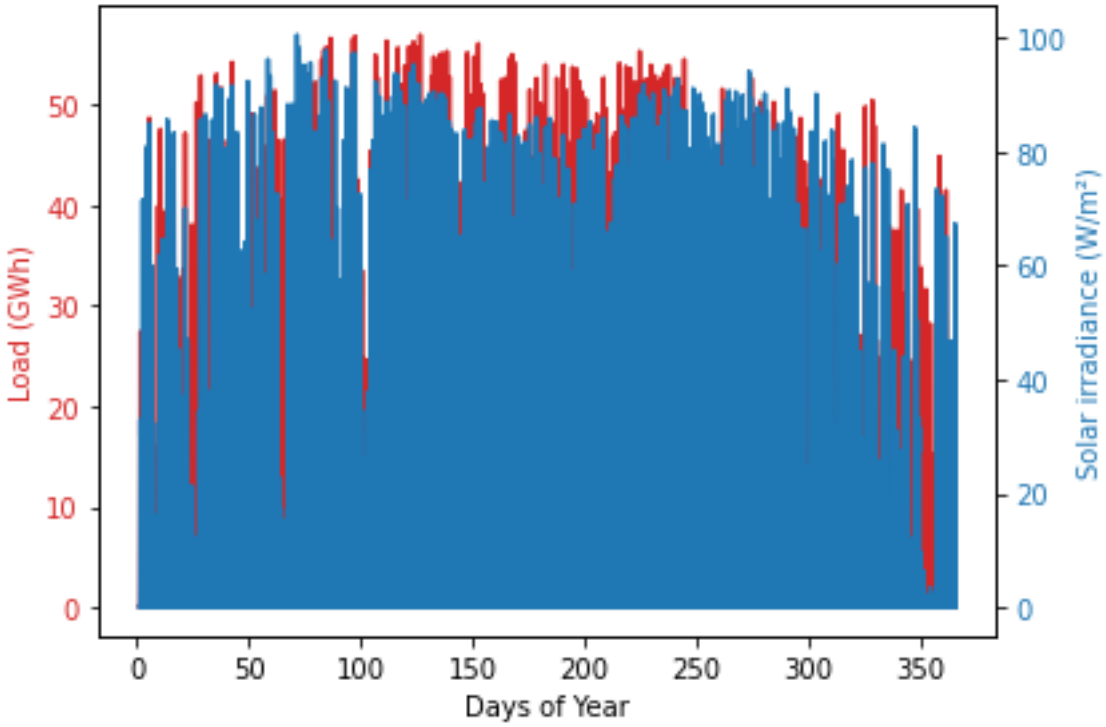


Figure 4.5 Correlation between PV Energy and Solar irradiance

random forest. These statistical models rely primarily on historical data to predict future time series. Therefore, the quantity and quality of historical data are essential for an accurate forecast. Nowadays, the deep learning model becomes more popular in solar irradiance forecasting. The deep learning model, which is the subpart of the machine learning model, was developed to solve a complex problem with a large amount of data. The multiple layers in the deep learning structure automatically learn the abstract features directly from the raw data to discover useful representations Deep learning models are distinctive from other machine learning models because they outperform as input data scale increases. Ng et al. [27] compared machine learning and deep learning models' performance while changing the amount of input data. The result showed that deep learning models tend to increase their accuracy as the number of training data increases, whereas the traditional machine learning models stop improving at a certain amount of data. The deep learning models specialized for handling sequential or time-series data such as text, speech, and image have been developed and have been successful. Recurrent neural network (RNN), long short-term memory (LSTM), gated recurrent unit (GRU), and convolutional neural network-LSTM (CNN–LSTM) models are typical. Because solar forecasting is intrinsically based on sequential data, such deep learning models were also applied for solar forecasting. For instance, Zang et al. [28] demonstrated that the accuracies of CNN–LSTM, LSTM, and

CNN models are better than those of ANN and SVM. As previously assumed, three models of neural networks are to be constructed. Given that in this case we are dealing exclusively with exogenous factors, and not with consumption and energy production, it is imperative to check the linear correlation between the possible inputs and the output which is PV produced energy. The relation between the generate power and each medal logical variable was studied. The correlation coefficient for two variables quantifies the dependency between them. It is mathematically given as:

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \hat{x})\,(y_i - \hat{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \hat{x})^2 \sum_{i=1}^{n}(y_i - \hat{y})^2}}$$

Where x_i&y_i are elements of the vector X and Y, respectively. These are the vectors with the values of the variables and the parameters $\hat{x}$ and y represent the arithmetic mean of the elements in this vectors.

A correlation coefficient near to one or minus one characterizes a high dependence. But as this confessions approaches to 0, the dependency starts to decrease. A new value defines the two variables have no relation to each other. The results are shown in table below, were only the 4 highest correlation coefficients were considered.

Input	Correlation
Direct Irradiance	0.874002
Diffuse Irradiance	0.718346
Temperature	0.769426
Barometric Pressure	-0.1424734

Table 4.1 Correlation between inputs and PV energy

One may inferred from the table that the climatic variables with the highest correlation with the solar generation are: direct irradiance, diffuse irradiance end temperature. all the parameters have a positive relation with photovoltaic power, that means that, as their value increases, the PV power also increases. In the other hand, barometric pressure has a negative relation with the PV power. This probably occurs because the greater barometric pressure is, the greater the humidity, so the greater the probability of there being clouds in the Sky and, consequently, of shading the photovoltaic system. Considering the correlation analysis, these three variables will compose the exogeneous input vector for our neural networks. Therefore, the neural networks in this case will use as input values:

i. 7 binary values corresponding to the day of the week

ii. 24 values of previous day temperature data

iii. 24 values of previous diffuse irradiance data

iv. 24 values of previous direct irradiance data

In the **Figure 4.6** below is depicted the developed network structure and data flow for the 3 possible neural networks.

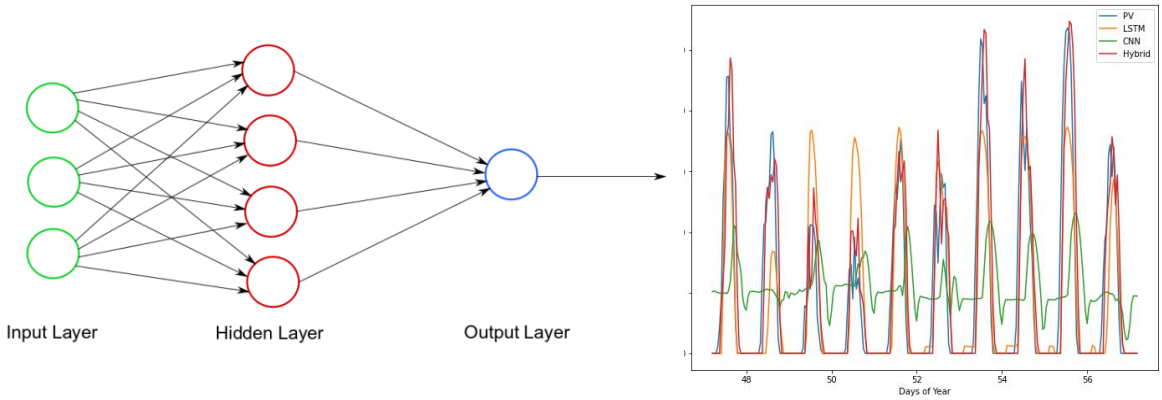


Figure 4.6 Developed network structure and data flow

4.1.3 Wind Energy Forecast

Wind energy is another one popular renewable energy source because it is unlimited source of wind which makes it environment friendly. A challenging issue associated with the Wind energy is that its power output strongly depends on uncertain and uncontrollable meteorological factors, such as wind speed and direction of the wind. Wind direction is a factor that can be eliminated with various changes in the design and operation of the wind turbine. As shown in the **Figure** below, there is a relative reproducibility in output and wind speed. Most models designed to predict wind energy had as input data both exogenous factors and the same wind power generation data, more will be mentioned below. Since the accuracy of NWP data has a very important effect on the accuracy of wind power prediction, one way to improve its performance is to reduce the uncertainty of NWPs. For this purpose, Kalman filtering algorithm is used to eliminate systematic errors. The Kalman filter as a group of mathematical equations presents the optimal estimation by merging last weighted observations to mitigate related biases. Below it is depicted the wind speed before and after the Kalman filter application. As in the case of solar energy, so in the case of wind energy, the external input factors will be determined by the linear correlation relationship. On the **Table 4.2** it can be seen the 4 highest correlation coefficients.

Input	Correlation
Wind Speed	0.803478
Wind Direction	0.148608
Temperature	-0.119301
Barometric Pressure	0.039248

Table 4.2 Correlation between inputs and Wind energy

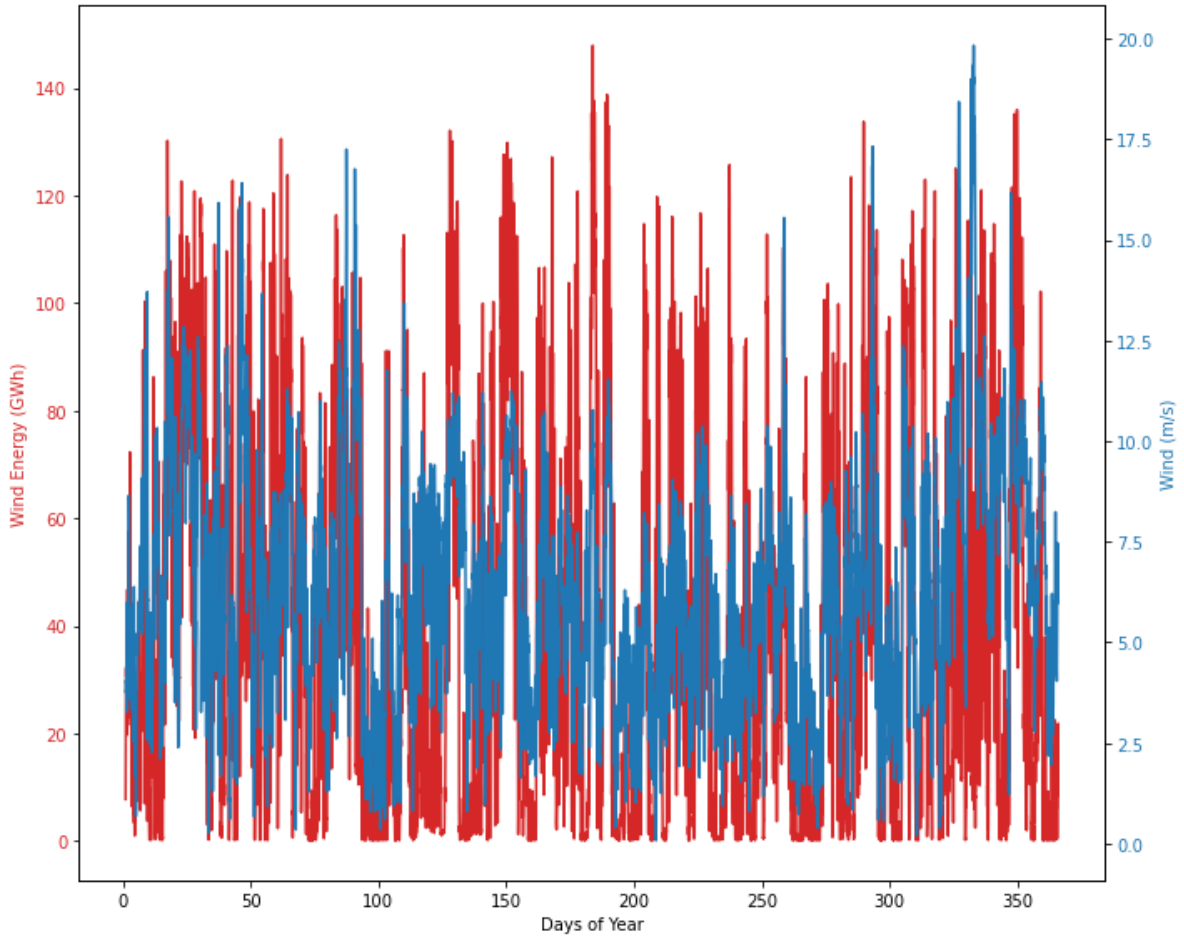


Figure 4.7 Correlation between Wind Energy and Wind speed

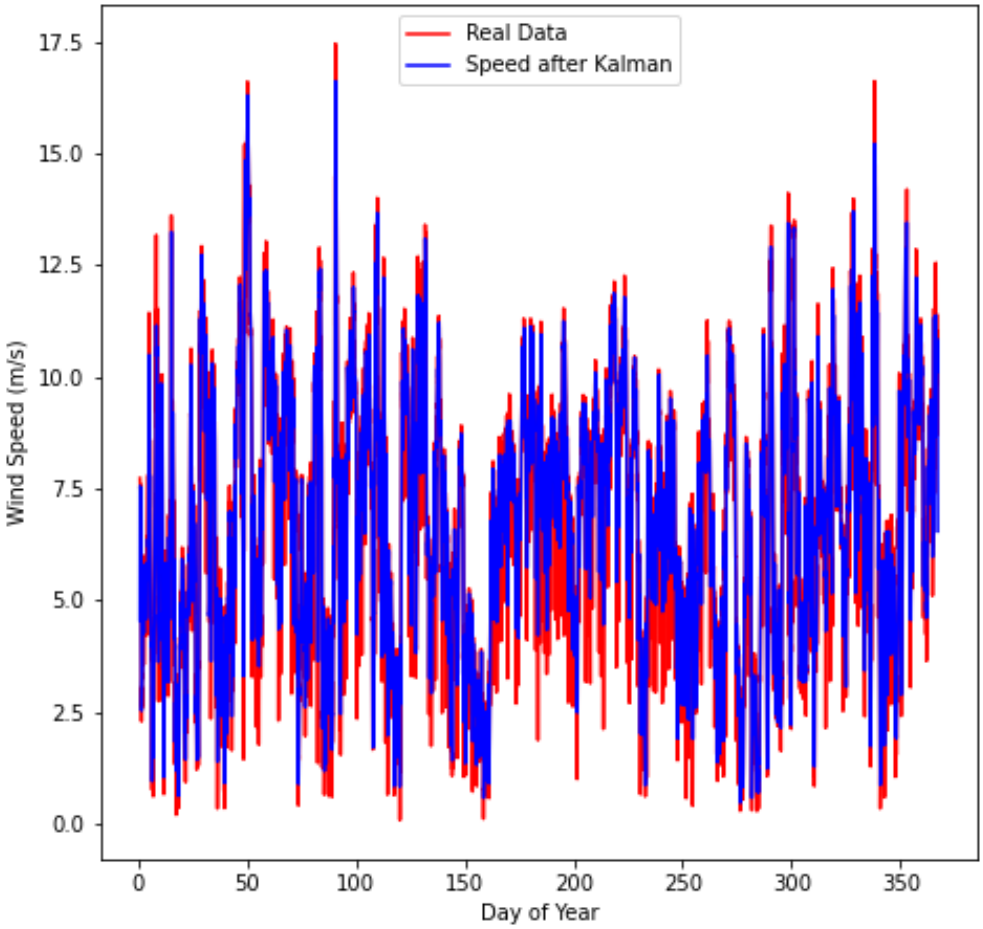


Figure 4.8 Wind speed before and after the application of Kalman filter

From the **Table 4.2** above, it is obvious that the main parameter that affects the produced by the wind generators is the speed of the wind. Therefore, the neural networks in this case will use as input values:

i. 7 binary values corresponding to the day of the week

ii. 24 values of previous wind speed data

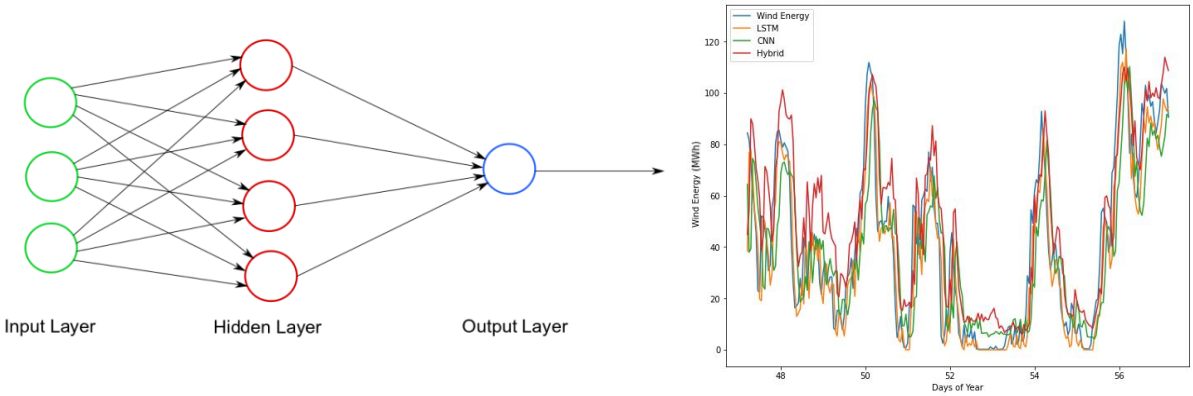


Figure 4.9 Developed network structure and data flow

4.1.4 Thermal Energy Forecast

Thermal energy is the energy produced by a thermoelectric generator by the combustion of either natural gas or some other liquid fuel. In this study case the power was being produced from the thermal energy generator is the 88% over needed energy. To make a proper algorithm which can handle the energy management system it is needed to the forecast all the energy sources including the thermal energy. below it is depicted the relation between thermal energy, the temperature and the energy demand of the island.

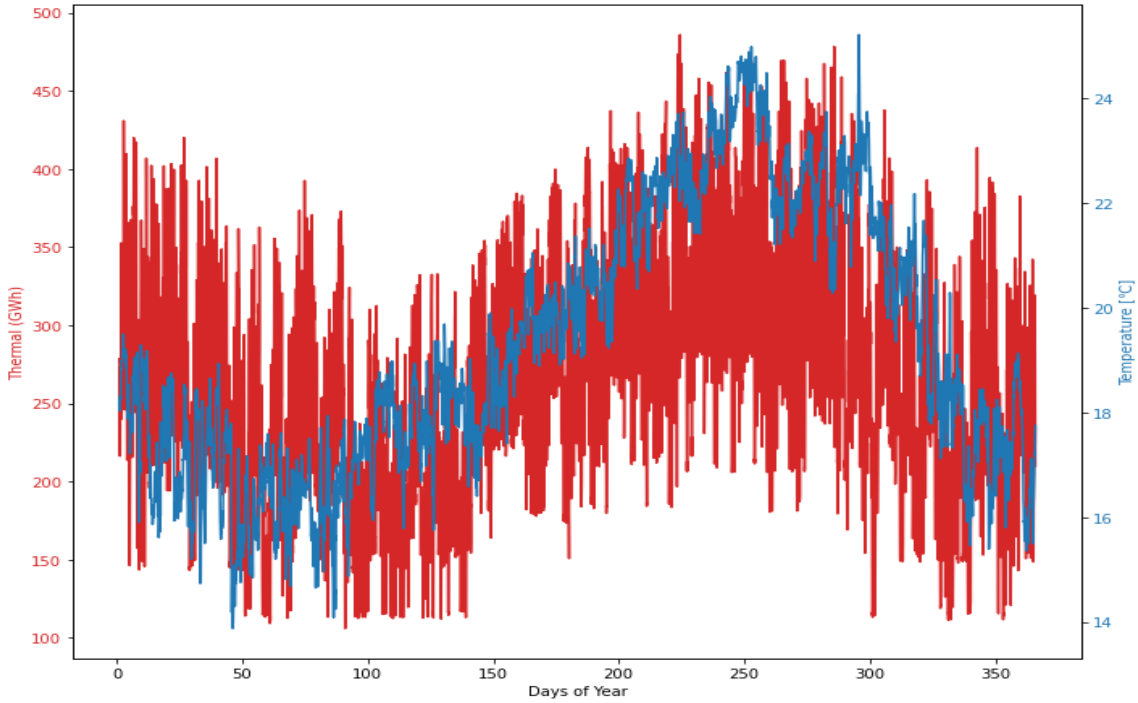


Figure 4.10 Graphic Correlation between Thermal Energy and Temperature

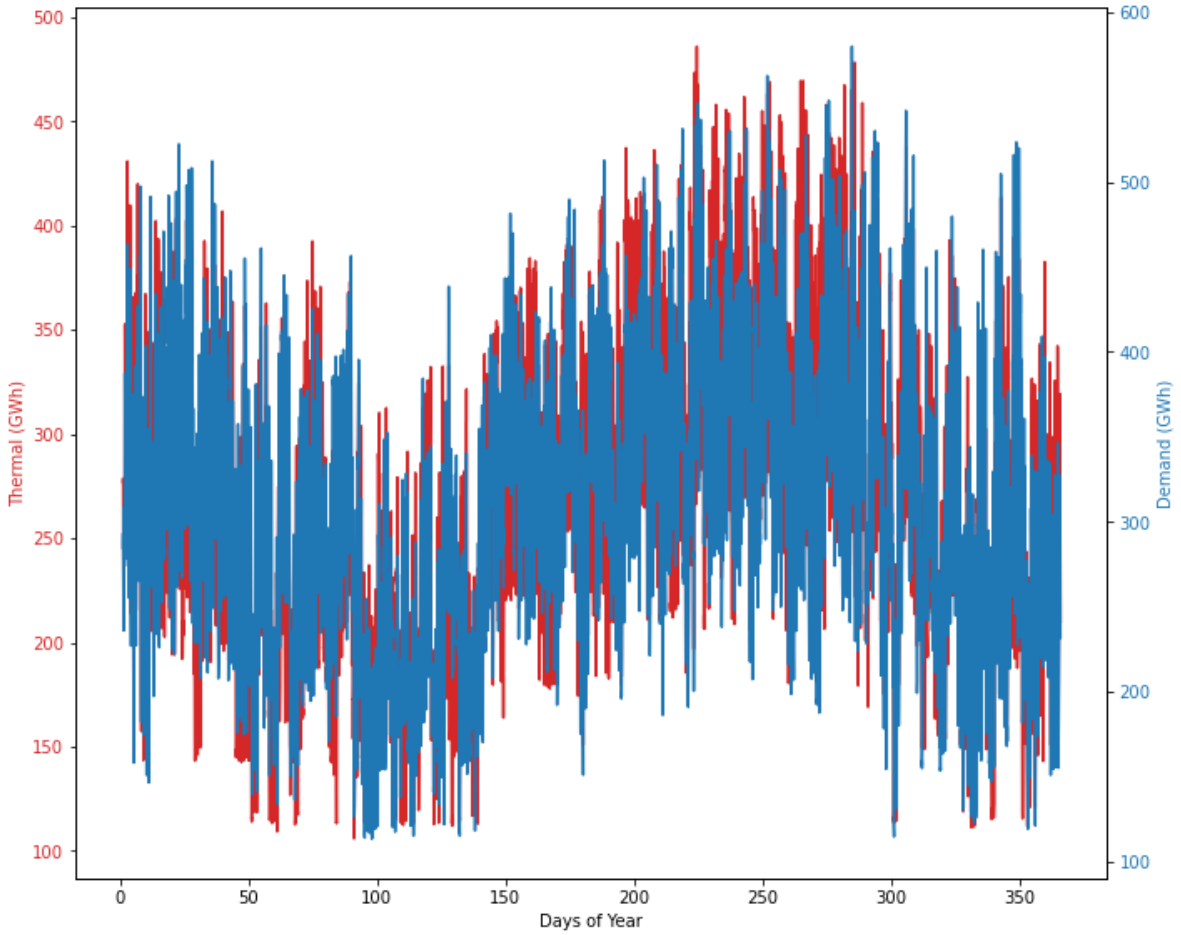


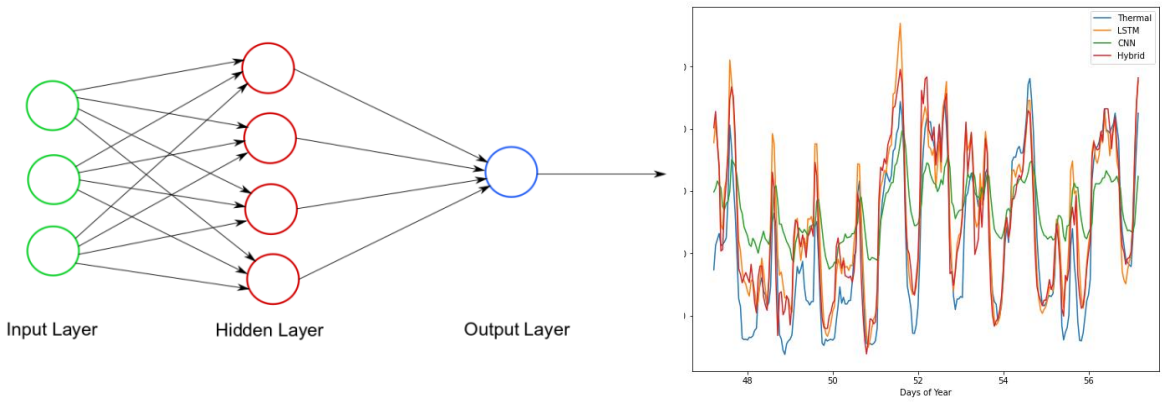
Figure 4.11 Graphic Correlation between Thermal Energy and Demand

As it is clearly, the thermal energy production follows the same course with the demand. Also, it cannot be observed a strong correlation between the trend of the thermal curve and the temperature data of the island. As a result of the above, then input values for the neural network chose to be:

i. 7 binary values corresponding to the day of the week

ii. 24 values of the hourly consumption data of the previous day,

iii. 24 values of previous day temperature data



On the figure below it is depicted the final neural network structure and the results of the forecasting.

Figure 4.12 Developed network structure and data flow

4.2 Performance Evaluation Metrics

This section compares the three models of all forecasts and discuss the experiments in context of depression detection in more detail. CNN, LSTM, and Hybrid are compared with each other, with the latter seems to perform most optimal. For this work, metrics were selected to evaluate the model. MSE (Mean Square Error), RMSE (Root Mean Square Error) and MAE (Mean Absolute Error). These metrics are defined as follows [30]:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}$$

$$MAE = \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)}{n}$$

All error metrics values of the three compared methods with all four energy datasets described below. On average, and most of the cases the proposed CNN-LSTM framework outperforms all other compared forecasting methods with reasonable computational time. The prediction results suggest that the deep learning methods are more suitable for volatile data description. Moreover, for MAPE, which measures the relative errors of the prediction results, the proposed CNN-LSTM framework shows lower error rates compared with all other methods for the three out of four forecasts.

4.2.1 Models Comparison and selection

4.2.1.1 Total Demand

On the table below it can be seen a comparison LSTM, CNN and Hybrid methods of forecast. The metrics that have been used for this comparison is MAE, RMSE and MSE. As it is clearly to see, the Hybrid model outperforms the other two models. Furthermore, the total needed time for this model was 120 seconds, when CNN needed 1148 seconds and LSTM 552.

Total Forecast	MAE	RMSE	MSE
LSTM	0.062	0.080	0.008
CNN	0.122	0.154	0.023
Hybrid	0.054	0.069	0.005

Table 4.3 Accuracy values comparison for three Neural Network Models

On the figure below it is depicted a graphical contrast between the four models and the given data. As is obvious the hybrid model succeeds better forecast results than the other models.

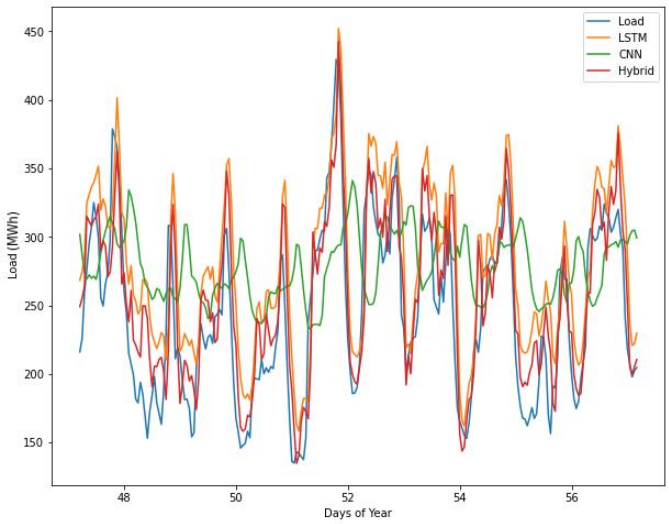


Figure 4.13 Results Compared for Total Demand

4.2.1.2 Solar Energy

Solar	MAE	RMSE	MSE
LSTM	0.089	0.143	0.020
CNN	0.195	0.269	0.072
Hybrid	0.039	0.068	0.005

Table 4.4 Accuracy values comparison for three Neural Network Models

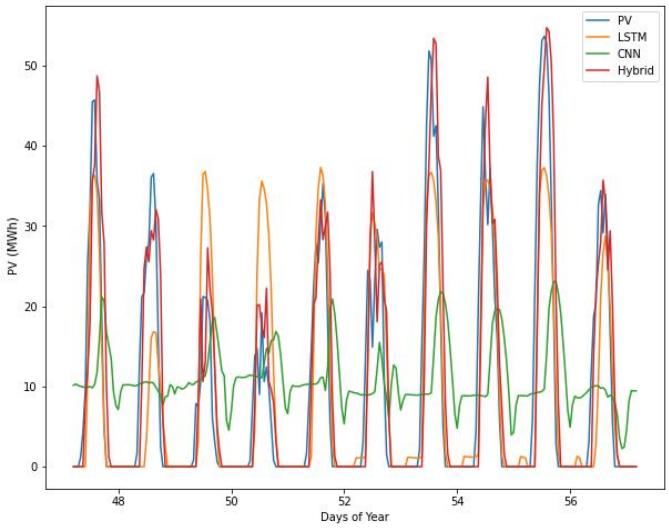


Figure 4.14 Results Compared for Solar Energy

As in the case of the full load and here it is easy to observe that the hybrid model is far better than the rest. However, what is observed is that compared to the LSTM model, the hybrid has almost two times less error values, in MAE and RMSE. This was not observed in the forecast of the total load. The reason this is probably the case is that in the case of solar energy, inputs are not energies but exogenous factors such as solar radiation.

4.2.1.3 Wind Energy

Wind	MAE	RMSE	MSE
LSTM	0.049	0.097	0.009
CNN	0.102	0.134	0.018
Hybrid	0.048	0.065	0.004

Table 4.5 Accuracy values comparison for three Neural Network Models

In the case of wind energy the same phenomenon is observed as in solar energy. The Hybrid model is the more accurate again.

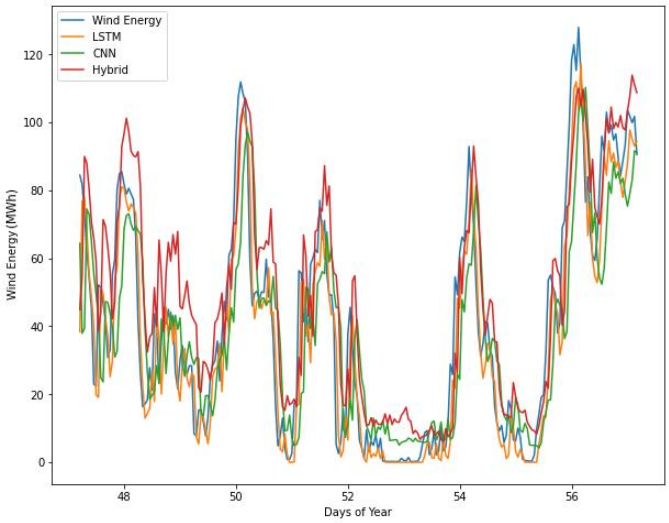


Figure 4.15 Results Compared For Wind Energy

4.2.1.4 Thermal Energy

Thermal	MAE	RMSE	MSE
LSTM	0.070	0.068	0.005
CNN	0.192	0.244	0.060
Hybrid	0.075	0.143	0.016

Table 4.6 Accuracy values comparison for three Neural Network Models

In the case of thermal energy, it is observed that the simplest LSTM model works better and results in lower errors. The results can be seen in the figure below.

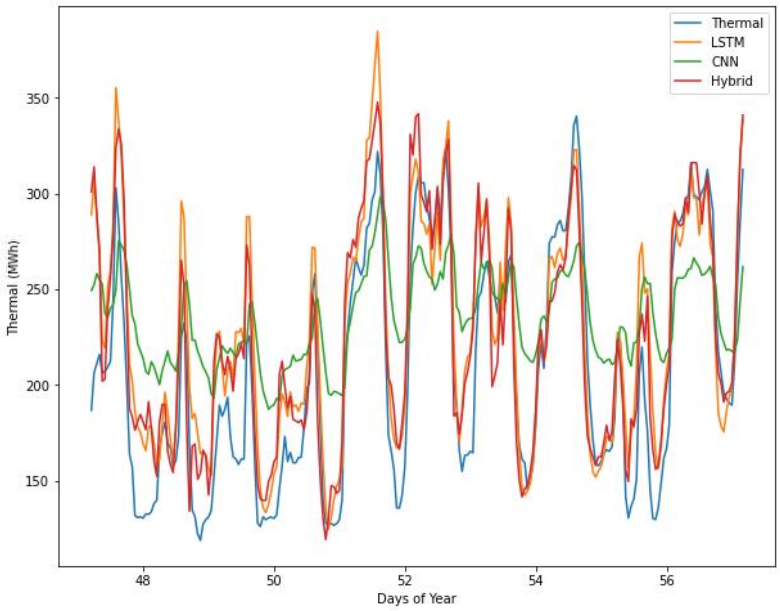


Figure 4.16 Results Compared for Thermal Energy

Summarizing the selected models used to predict total, wind, solar and thermal energy is a Hybrid for the first three and LSTM for the latest.

5. Optimization Algorithm

In this chapter, an algorithm was developed to be integrated into the energy management system of the islandic power network, that could upgrade its operation by providing renewable energy in a more secure and stable way and by peak shaving the maximum demand value. The developed algorithm was suited for the specific island case investigated in this study but could be also implemented with few modifications to other similar power systems, where the load curve presents similar patterns. In the following section, the algorithm is described in detail.

5.1 General Problem

After the load forecasting module was configured and tested, the demand values of the next day at an hourly basis could be predicted. This ability was taken into account for the development of a predictive Energy Management System (EMS) algorithm that could be used for the "smartening" of the power system, considering a more stable and robust operation. The facts that the load curve's most frequent shape and pattern was characterized by high peaked values at morning hours and by highly peaked values at late night hours and that the type of installed renewable sources was PV, were considered for the decision of the operational modes of the developed algorithm. Another main problem that should be solved was the unpredictable balances between load and generation from Wind Farm.

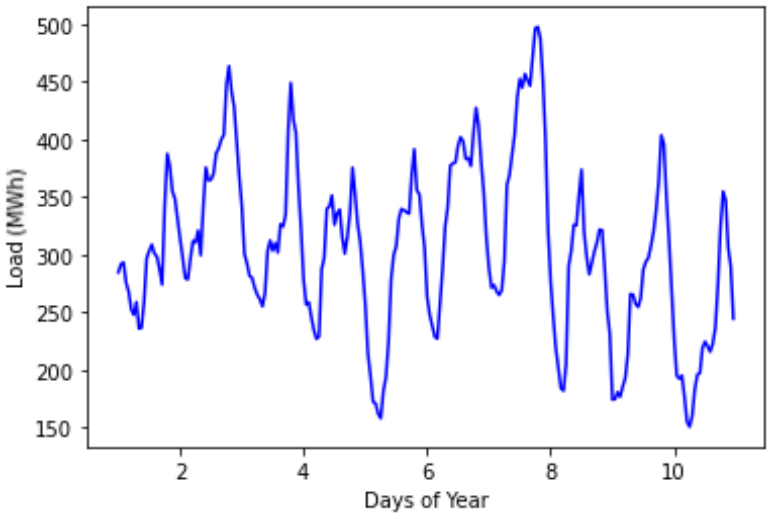


Figure 5.1 Demand curve in the first 10 days of the Year 2016

From this figure, it can be proved that there are two demand spikes for most of the year. The first demand spike is detected during the morning hours where everyone wakes up and starts consuming energy and the other is detected during the evening hours where most people return from work and once again they begin to consume energy in their homes.

Considering the shape of the load profile, it can be concluded that especially for the winter period, the peaked values of load are detected during the night hours where the renewable installed PV power is not available. From figure above it can be shown that the island's power system has maximum peak demand values about 500 MW, which are currently covered by conventional diesel generator in combination with Wind Generator. The installed capacity of Wing Turbine is approximately 40MW accounting for around 13% of the peak values, on the other hand PV power reaches 11MW witch computes to 3.5% of the total energy. Regarding the above, it is rational to consider the following targets be countered by the algorithm:

1. Peak shaving of the maximum demand values of each day of the year
2. Smoothing the operation of the thermal engine during the off-peak hours
3. Smoothing the load curve of the Wind turbine
4. Avoiding the "duck" shape evolution to the load curve, concerning the constantly increasing PV power penetration to the grid
5. Ensuring 100% renewable energy penetration

Concerning the point 4, the "duck" shape is presented in the following figure (Figure 21) and is a trend that is constantly evolving and affecting the net load power curve. The net load power curve is defined as the total load curve subtracting the PV power production curve. The resulting curve (which is named net load curve) is the one that will have to be covered by the rest thermal-conventional power production units. As more and more solar PV are integrated into the grid, it starts dramatically suppressing net load during midday, when the sun is out. The net load curve sags in the middle of the day (like a belly) and then swoops back up when the sun goes down (like a neck).

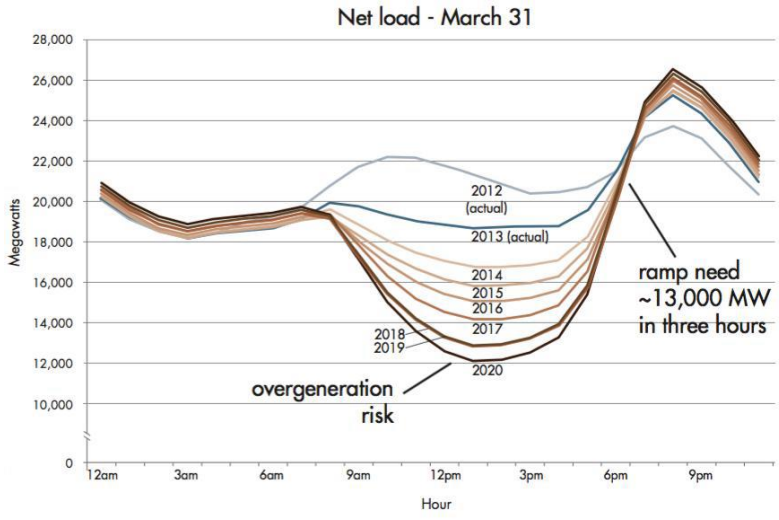


Figure 5.2 The “duck” net load curve, illustrating steep ramp needs and overgeneration risk [38]

The figure above was taken from the system service operator of California [38] and the “duck” shape evolving through the years from the increasing PV power generation is depicted. It is also clear that this effect is accompanied by an overgeneration risk which is due to the technical minimum operating point of the thermal units. The “duck” shape is also directly related to the need for higher ramp rates because of the sudden load change. These effects will have a negative impact on the stable and smart operation of a future isolated grid with high renewable penetration. The main idea for encountering the above effects and achieving a smoother, robust system operation was to take advantage of the load forecasting ability and to combine it with an energy storage system (ESS) in order to save a specific amount of energy that could be injected to the power grid later. For the purpose of this study and based on the fact that battery energy storage systems (BESS) have recently made a resonant entrance to the power market and are constantly evolving, such a system was considered for the storage ability needed in our case study.

5.2 Wind Energy problem description

Integration of renewable energy resources, and especially wind generated as it is easily spotted on the **Figure** below, to a power system can cause power fluctuations due to their intermittent nature.

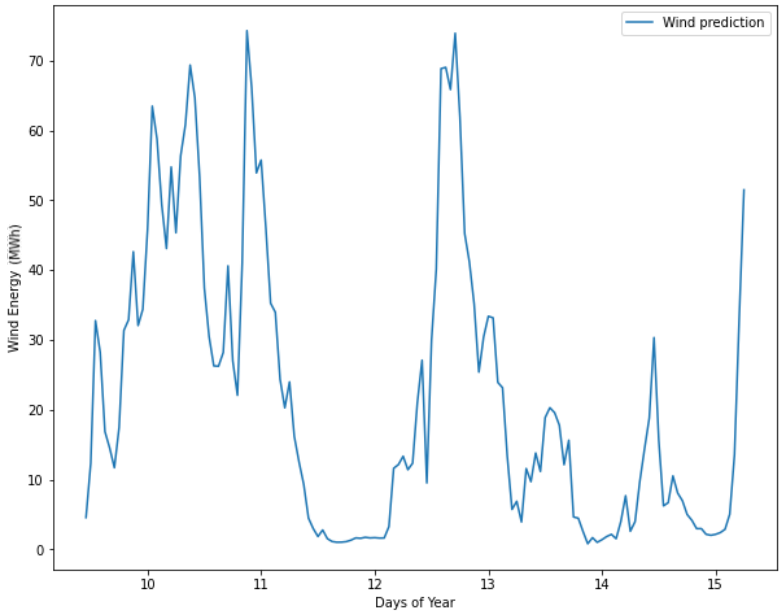


Figure 5.3 Wind predicted energy curve

In order to use wind power energy in the electricity market despite of its fluctuation, that makes it impossible to connect the power directly to the grid, smoothing the wind power output is one of the best possible solutions that can be applied to resolve this problem. There are many studies on how to stabilize output power for wind farm. Smoothing wind power fluctuations by controlling the pitch angle is a proposal in [31]. Although this control strategy can smooth the output power of wind farm, pitch angle is not responding fast when output power of wind farm need to adjust quickly, due to pitch angle can be adjusted 7 degrees per second. So the wind farm needs some transition time. Therefore, the output power of wind farm cannot track the power reference quickly. Uehera et al. [32] proposed an output power smoothing method by a simple coordinated control of DC-link voltage and pitch angle of a wind energy conversion system. Although the output power fluctuations of the wind power in the low and high frequency domains are smoothed by the pitch angle control and the DC-link voltage control, the DC-link voltage will have a large fluctuation. The large fluctuation is not conductive to wind power system stabilization. Mohammad et al. [33] proposed the method for smoothing the output power of wind farm by using the energy storage system, where is the method applied in this diploma thesis.

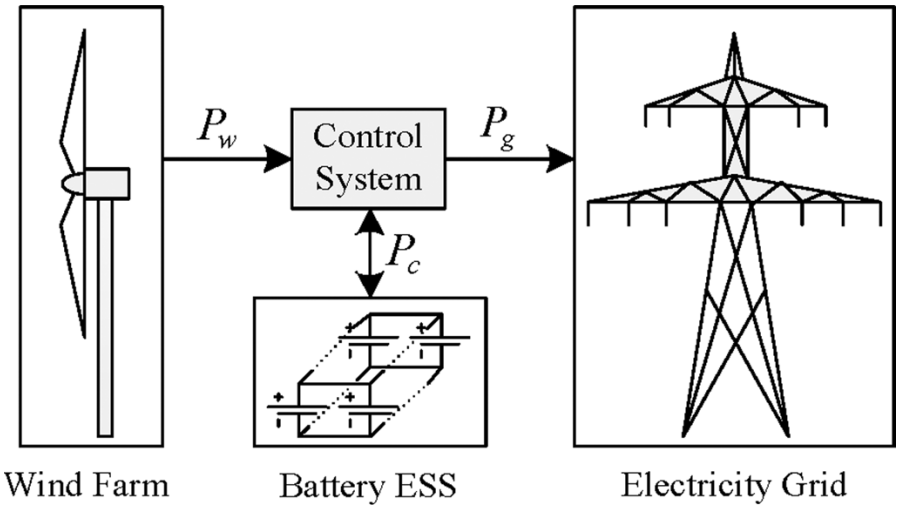


Figure 5.4 Wind BESS simulation

5.3 Wind Energy algorithm

As mentioned earlier, there is a need of smoothing wind energy curve. It's been considered a scenario where the energy supply side of the microgrid has a combination of wind turbines and battery. **Figure** below shows a conceptual schema of wind and battery hybrid system. based on the figure a smoothing algorithm is applied to smooth out power fluctuations. After that, the controller of the system is going to be built will be working with a specific algorithm. this algorithm has as a goal to store electric energy, produced by wind generators, in the battery energy storage system when the produced energy is more than the predetermined

and foreseen Wind power curve. In case the produced energy is less than the predetermined and foreseen the controller have to pull energy from the battery pack, to stabilize the less production. The final smoothed power is expressed as:

$$P_{sm} = \begin{cases} P_{wind} + P_{bat}^{d} \\ P_{wind} - P_{bat}^{C} \end{cases}$$

, where P_{sm} stands for the smoothed power, P_{wind} for produced energy buy wind generators and P_{bat} for energy managed by battery. In case of adding energies is discharging and in case of energy substruction the battery is charging. Moving average is a technique to get another idea of trains in a data set. the averaging technique it's based on a sliding window size. the type of moving average which being used in this thesis, for smoothing data is the simple moving average also known as SMA. According to Alessio et al. [34], a moving average method is a well-known low-pass filter for time series and it is defined as:

$$\tilde{y}(i) = \frac{1}{w}\sum_{k=0}^{w-1} y(i - k)$$

where $\tilde{y}(i)$ is a time series data with window length w. Although it is simple and traditionally accepted way to reduce fluctuations in renewable resources, it exhibits a memory effect which depends on the length of the averaging window. Numerically, a moving average with window length of w contains only $(\frac{1}{w})$% of present values of a fluctuating variable.

5.3.1.1 Measure of smoothness

An expression for common mathematical definition of smoothness is written as [Kenneth J Adams and Donald R Van Deventer. “Fitting yield curves and forward rate curves with maximum smoothness”. In: The Journal of Fixed Income 4.1 (1994), pages 52–62.]:

$$Z = \int_{0}^{T} [f''(s)]^2 \, ds$$

for function $f(t)$, where $t \in [0, \ldots, T]$. Equation ___ is described as an integral of squared second-order differential of function $f(t)$. The minimum value of Z corresponds to the maximum smoothness level.

5.3.2 Determining battery size

To determine the maximum capacity of the battery energy storage system which being used for the smoothness off the wind power curve is needed to consider charging and discharging cases and then take the maximum of these cases. In the **Figure** below is depicted a diagram which shows the relation between the predicted and the smoothed power curve.

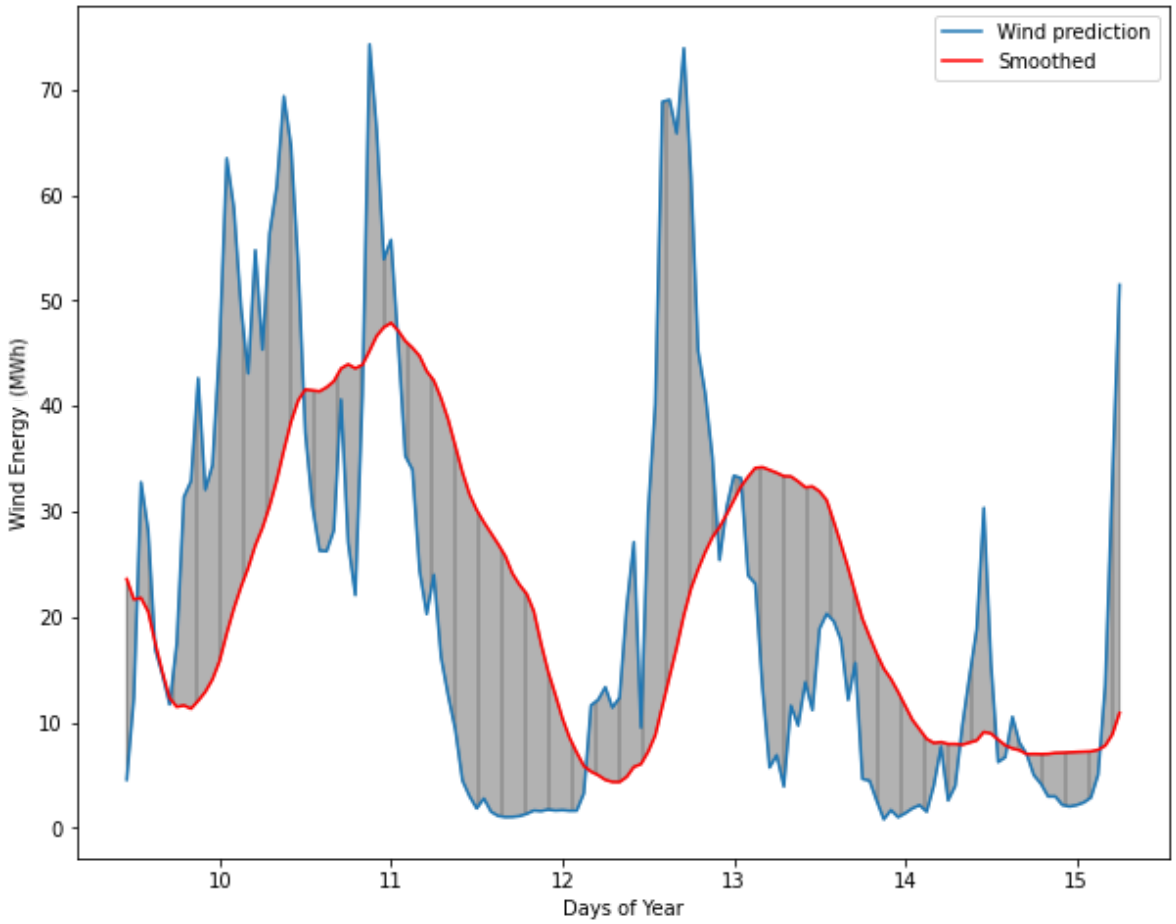


Figure 5.5 Correlation between Predicted and Smoothed power curve

The areas above the smoothed curve represent the over produced power and the areas below the need of extra power. The calculation of them will be presented analytically below.

5.3.2.1 Charge operation

The amount of power to be stored in the bess is represented by the area above the smooth power curve and below the predicted curve. Therefore, the amount of charge is determined by the following equation:

$$E_C = \int_0^T [P_{pr}(t) - P_{sm}(t)]dt$$

where $t \in [0, \ldots, T], P_{pr}\ and\ P_{sm}$ represent predicted and smoothed power, respectively, each time.

5.3.2.2 Discharge operation

The amount of power to be absorbed from the bess is represented by the area below the smooth power curve and over the predicted curve. Therefore, the amount of discharge is determined by the following equation:

$$E_D = \int_0^T [\, P_{sm}(t) \, - \, P_{pr}(t)]dt$$

where $t \in [0, \ldots, T], P_{pr}\ and\ P_{sm}$ represent predicted and smoothed power, respectively, each time.

5.3.2.3 Needed computed capacity

To compute the required capacity it is necessary to compile a smart algorithm for calculating this. The obvious condition is that the needed capacity has to be at least equal to the minimum value of the discharging energy or the maximum value of charging energy. For example, if the absolute value of discharging energy is larger than the absolute value of charging energy, this is the minimum capacity off the battery pack. This capacity is sufficient in case that the total charging amount is less than the calculated capacity. At first, the algorithm that is responsible for the calculation of needed capacity adds up in pairs, same situation the calculated areas, until it reaches in a sequence of positive and negative numbers where they alternate. Now the charging capacity is the $maxE_C$ and the discharging capacity is the $max|E_D|$. The optimal battery capacity was described and calculated in [35]. In this research reference is made to battery functional characteristics and in particular charge and discharge efficiency, $\eta_c and\ \eta_d$ respectively. A similar equation with them, is the following:

$$E_{BESS} \, = \, max(E_C \, \eta_c, E_D/\eta_d)$$

This result is a bigger battery size than the originally calculated. The battery is considered as a Li-on battery. According to Chen et al. [35] a common value of $\eta_c\ and\ \eta_d$ for this type of batteries is set to 85% respectively. Using this equation the necessary capacity is 780MWh.

5.3.3 Proof of solution viability

Once all the parameters have been calculated, it remains to be proven that this solution is viable. As previously assumed, the wind energy curve is characterized by large fluctuations during the day. which leads to large fluctuations in the load produced by the thermal power unit. Creating a smoothing profile of the wind energy curve also results in the smoothing of the thermal curve. To determine if this smoothing has an effect and does not increase consumption, it is enough to calculate the a represent energy production, both in the non-smoothed and in the smoothed curves. The equation that is been used to calculate the total MWh is the following:

$$Total\ MWh = \int_0^T [P_{pr}(t)\ ]dt$$

where $P_{pr}(t)$ stands for the produced energy, either from the Wind or the Fuel and $t \in [0, \dots, T]$. Applying the above equation the following results are obtained:

Type of Energy	Non-Smoothed	Smoothed
Wind (MWh)	12399.01	12396.65
Thermal (MWh)	92907.36	92910.98

Table 5.1 Wind and Thermal total MWh before and after the smoothness for two weeks

At first sight, it can be observed that the energies produced are almost at the same levels before and after the application of the wind energy curve smoothing. However, smoothing the thermal curve will help build the algorithm for the overall energy storage system. In addition, it helps in other areas. It reduces the intermittency of wind power, which is called 'Ramp Rate' [36]. This phenomenon should be under control that this power be able to match the standards of electricity network grid.

5.4 Peak Shaving Algorithm Description

The idea behind of the second optimization algorithm with the use of a battery storage system it is very different from the previous one. In this time, the algorithm has the responsibility to save energy in the morning and use it in the night hours. In that way the peak demands in the night will be shaved.

The first step of the algorithmic procedure, was to determine the inputs. Those were decided to be the 24 variables vector, which contained the demand values of the next day, as they were predicted from the ANN model and the 24 variables vector containing the hourly values of the total PV energy production of the corresponding day. Then the new peak level was decided and set to be at 350 MW value of each day, for the thermal engine. This value was determined based on the load curve data observation and was evaluated from the values of the peak region relatively to the mid-day load values. However, it is essential that there is a standard amount of charge on the batteries or that you shave every day. For this reason, the algorithm developed may reduce the maximum value each day, but in some cases it is kept constant in order to achieve additional charging of the battery system. After the peak reduction level or peak shaving level was determined, based on the load forecasting of the next day, the area to be removed after the peak reduction was calculated. This is better visualized in the following **Figure 5.6** . In this figure, the load curve of the next day is depicted with the solid line and the peak reduction level that is decided to be implemented, is represented with the dashed line. This line should cross the load curve at two separate points and in this way a closed area is shaped. This region, which is basically the area under

the load curve subtracting the area under the dashed line and is depicted as the grey region in **Figure 5.6**, above the red and below blue line, corresponds to the total energy that will be eliminated after the algorithm operation and the generation of the new diesel engines setpoints. It is also clear from the units of the two axes of **Figure5.6** that the a forementioned area, represents the energy values based on the following definition:

$$E_{1\to 2} = \int_0^T [P_{sh}(t)\,]dt$$

where $t \in [0,\ldots,T], P_{sh}\ shaved$ power each time.

With the above input parameters defined, the next step was the creation of the combined curve. For this, the "offset" value concept was necessary. The offset value was defined as a constant load value and at the first step of the algorithm it was initially set equal to the base load value, defined as the minimum of the daily load curve. This, offset value was used for the initial combined curve generation, which was later updated during the iterations of the algorithm. The combined curve was defined as i) the summation of the load curve and the PV power curve for the load values that were smaller than the offset value and ii) as the summation of the PV power curve with the offset value, for the load values that were greater than the offset value.

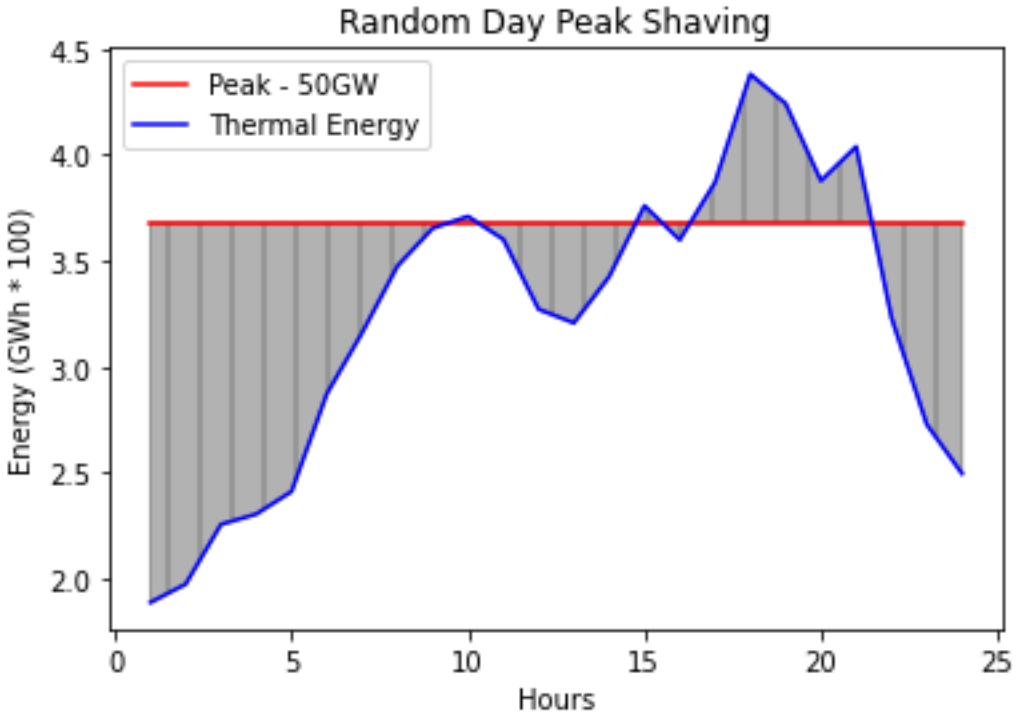


Figure 5.6 Operation of peak shaving algorithm in a random day

This is also mathematically described as below:

$$P_{final} = \begin{cases} P_{offset} + P_{V,} & if\ P_{offset} \geq P_{load} \\ P_{load} + P_{V}, & if\ P_{offset} < P_{load} \end{cases}$$

Observing the graphs of the whole year, two types of curves are observed. The first has two peaks, one peaks in the morning and one larger in the evening. The second form of the curve is in the shape of an inverted "U". This is observed only in the summer period where the consumption is increased throughout the day, since it is a period of holidays and intense tourist traffic, therefore high loads are required throughout the day.

5.4.1 Winter Period Example

As it mentioned earlier, the pattern of the graph is related with a period. in the graph below, it is depicted a winter day, more specific the 6th of January.

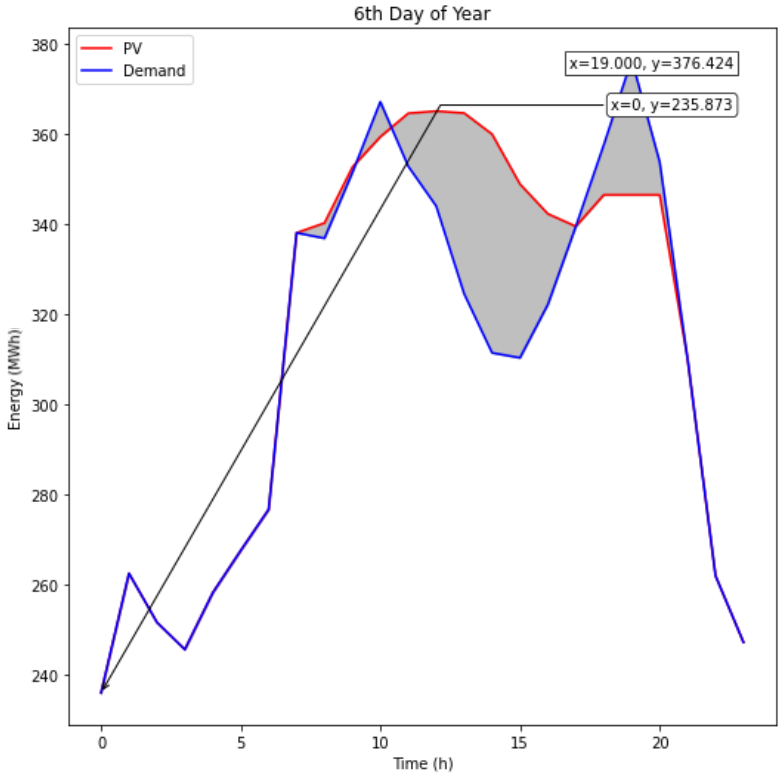


Figure 5.7 Comparison of Final Curve with Demand

In the above figure, the combined curve that the developed algorithm prescribed, is directly recognizable as the red curve. It is also possible to observe the high and low limits of the offset value. These are the lowest and highest demand values of the day, which define the range of possible values that the offset variable could gain. For that specific day of the year, a relatively smooth PV production curve accompanied by a smooth load curve can be

noticed, which is mainly characterized by its peaked values late at the evening. Another peak value can be spotted in the start of the day. PV production is maximised between 9:00 AM and 11 AM. between 8:00 and 10:00 o'clock and from 12 to 16 o'clock an artificially energy surplus can be spotted. This surplus can be stored in a battery to use it for night peak shaving or as a fuel for electric vehicles. From the above figure, it is possible to observe the area between the load curve and the offset value, as defined from the developed algorithm for a typical winter day. The red area represents the amount of energy that will be needed to be covered from the current PV energy production, assuming a perfect forecast for the load curve of the next day. This assumption is based on the fact that given the load curve of the next day and the PV power generation, the developed algorithm estimates the offset value, which is interpreted from the diesel engine operation perspective, as the maximum allowed value, regarding the time period. Thus, the red area of **Figure 5.8 a**, which is generated for every day with a similar winter demand profile, is generally partially covered from the current PV power production or even from stored energy in the battery system. In case neither of these power sources is capable of matching the demand of this area, a violation of the upper barrier posed for the diesel engine from the offset value is necessary, so that the system balance is preserved. In **Figure 5.8 b**, two regions are depicted. The orange region which corresponds to the red area above and the yellow region which corresponds to the excess energy area to be stored for later peak shaving capability. The orange area is actually the overlapping of the combined curve with the load curve and the offset line barrier whereas the yellow area is the net surplus energy and corresponds to the section

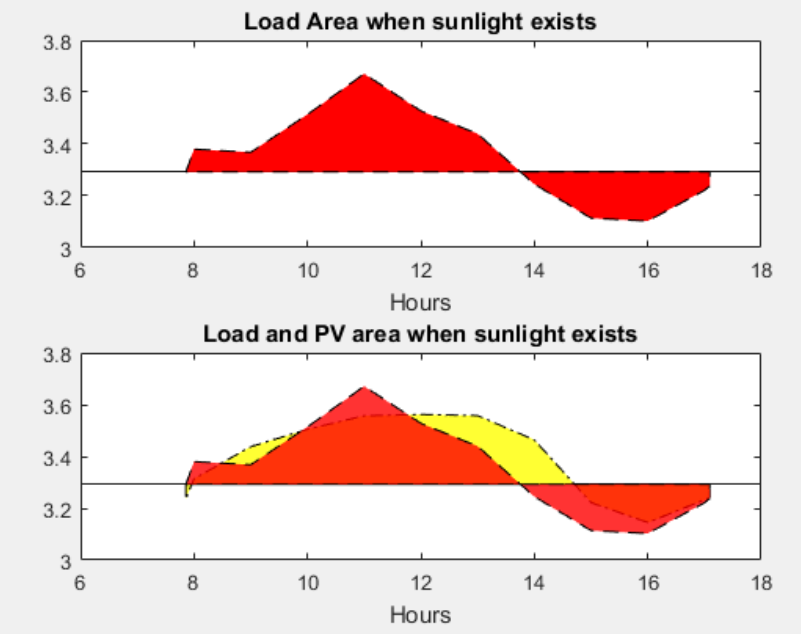


Figure 5.8 Artificially created excess energy and the corresponding demand, for day 6

PV line which is above the Load Curve. As it is possible to notice, the load curve and they PV curve ends at about 17:00. This fact is attributed to the transition of the combined curve to values lower than the current demand values and since the excess energy to be stored has to be renewable. From the following figure, play thermal energy planning is possible to observe. the new thermal plan is depicted with the red line and the old thermal curve with the green. The blue line at the same plot, represents the load curve. The green line represents V operation of the thermal engine when it used to inject the whole PV generated power in the island's grid. As this may be possible for a flexible small-scale power grid, this is not the case for a large power grid which is interconnected with many conventional thermal power plants. This is due to the achieved technical minimum operation that is prescribed for these units and the related start-up and shut-down dynamic operation modes. The last, are related with rapid and abrupt changes in the loading state of the thermal units which are subject to technical limitations. This in turn, may result in curtailment issues and valuable renewable energy is not injected into the grid, for stability reasons. This drawback is countered by the developed algorithm, as long as a stable diesel operation with predefined ramp-ups is achieved. At the same time, the renewable power production is ensured and smoothly integrated with the diesel and BESS operation.

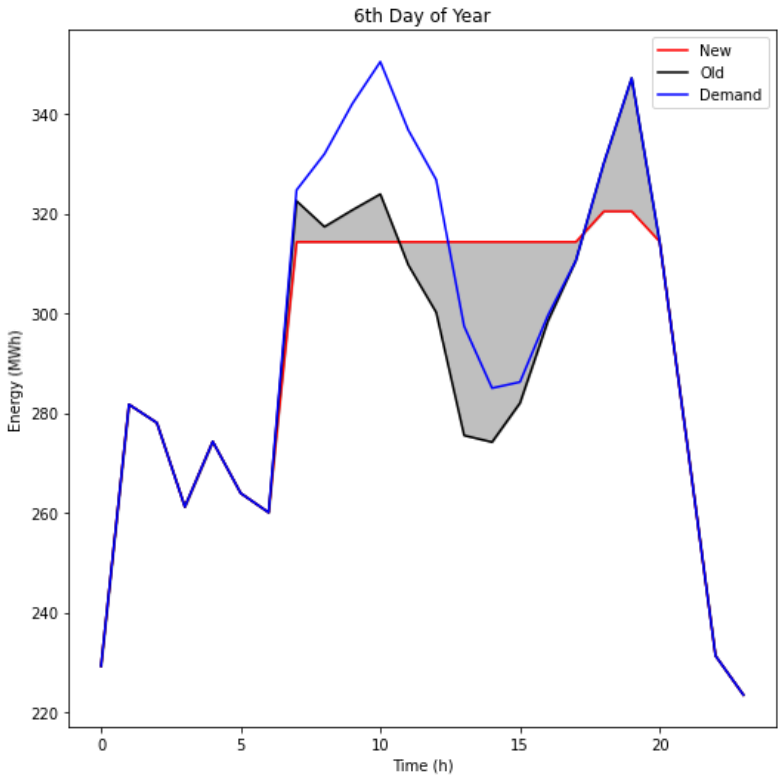


Figure 5.9 Algorithm output curve for 6th day

5.4.2 Summer Period Example

In the section above, the operation of the developed algorithm is described for the load curve pattern which are more frequently observed during the winter time period where the peak demand is displaced at the evening hours where there is not PV power production. Though this is the case for the winter time period for the specific islandic power system under investigation, the same pattern is also observed during the summer time period for many islandic power systems. For small-scale systems in which the load is commonly shaped by the residential factor and the touristic activities, the peak demand is constantly placed at the night hours, when the majority of those activities requires energy consumption. Nevertheless, in order to observe the behavior of the algorithm during the summer time period for the power system investigated in this study, the algorithm was executed for a whole year period, including each day of the year.

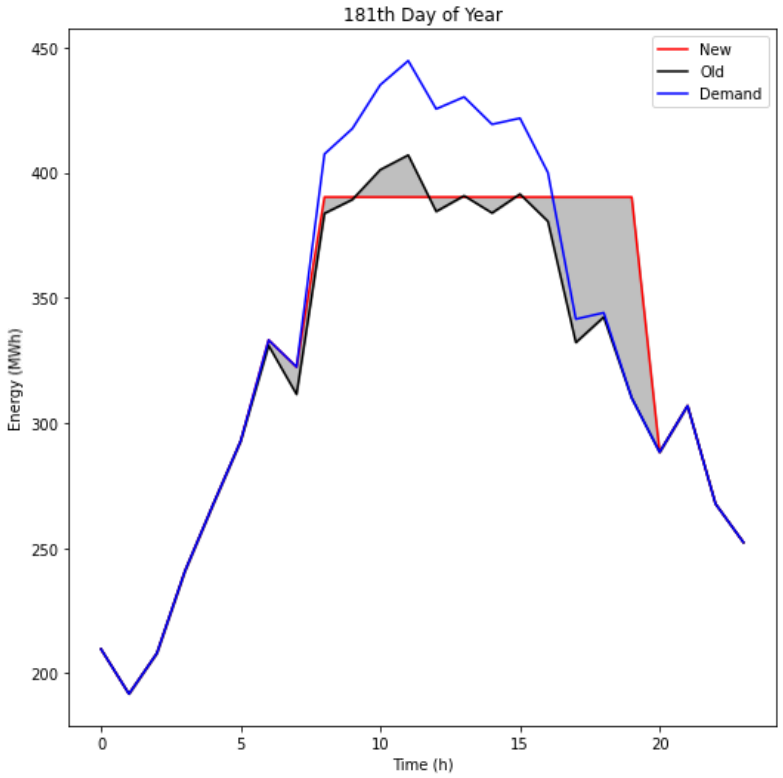


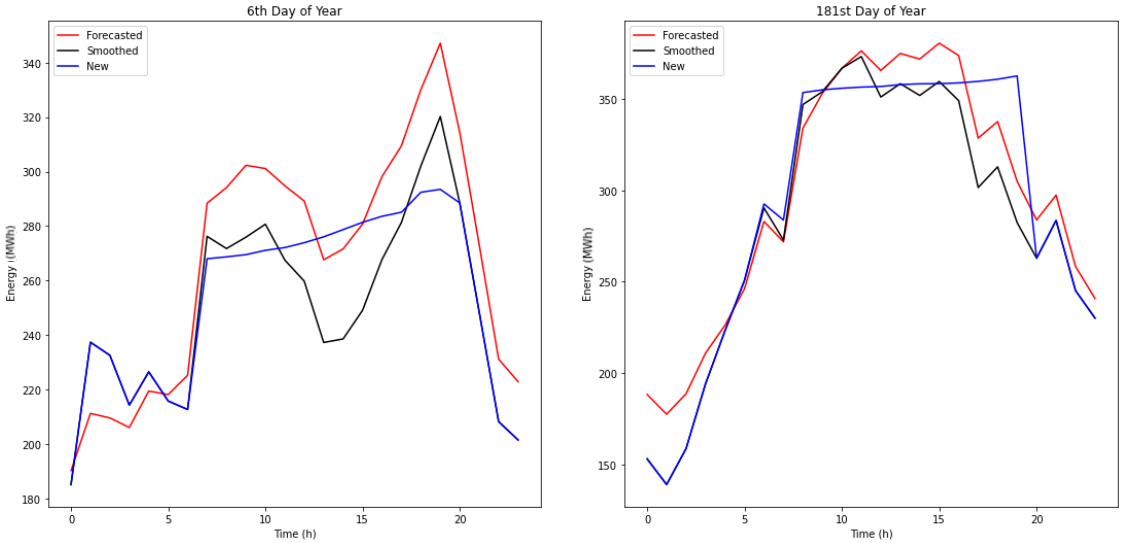
Figure 5.10 Algorithm output curve for 181st day

As it is obvious that load curve is very different from the curve in the winter. In summer there is an increase until the middle of the day, it keeps a steady pace for some hours and then it drops. It is a diverted “U”. On some days this pattern may not be observed but there is a very small drop in the required energy at noon and then again an increase in energy consumption. This happens because of tourism period and more movement in the island on the midday. There is a small peak 11:00 o'clock but the drop from that is almost nonexistent.

In the algorithm that was built is able to charge the batteries in case there is a large amount of energy needed the next day. This option was made to have a standard quantity of energy stored in the batteries every day, having as its future work the Vehicle to Grid.

5.5 Final Results

The above algorithms are designed to normalize the power generation curve of the thermal engine, to reduce the output from it as much as possible, to introduce renewable energy



sources as much as possible and to reduce the peaks of the thermal engine. From all the above actions three curves have emerged, the Forecasted thermal curve, the Smoothed and the New curve. In the figure below, it is depicted a comparison between these type of curves on the 6th and the 181st day of year.

Figure 5.11 Comparison of two curves

At first sight it is obvious that there is a considerable reduction in energy production by the thermal engine comparing the Forecasted output with the Smoothed. In addition, it is quite obvious that there is a smoothness of the curve. In particular there are no longer the sharp fluctuations from hour to hour due to the abrupt change of produced wind energy. The comparison of new curve with smooth was done before. The following tables show the difference in peak gigawatt hours produced during these days.

6th day comparison		
Forecasted (MWh)	Smoothed (MWh)	New (MWh)
350	320	300

Table 5.2 6th day peak energy comparison

181st day comparison		
Forecasted (MWh)	Smoothed (MWh)	New (MWh)
375	365	350

Table 5.3 181st day peak energy comparison

Initially there is a large decrease in output from phase 1 to phase 2. As well as an increase in phase 3 which is also less than phase 1. This is done as the energy which is produced by fuels it is used to charge the batteries extra when necessary. On the figure and table below it is depicted the 196th day of the year, when no extra energy it is needed.

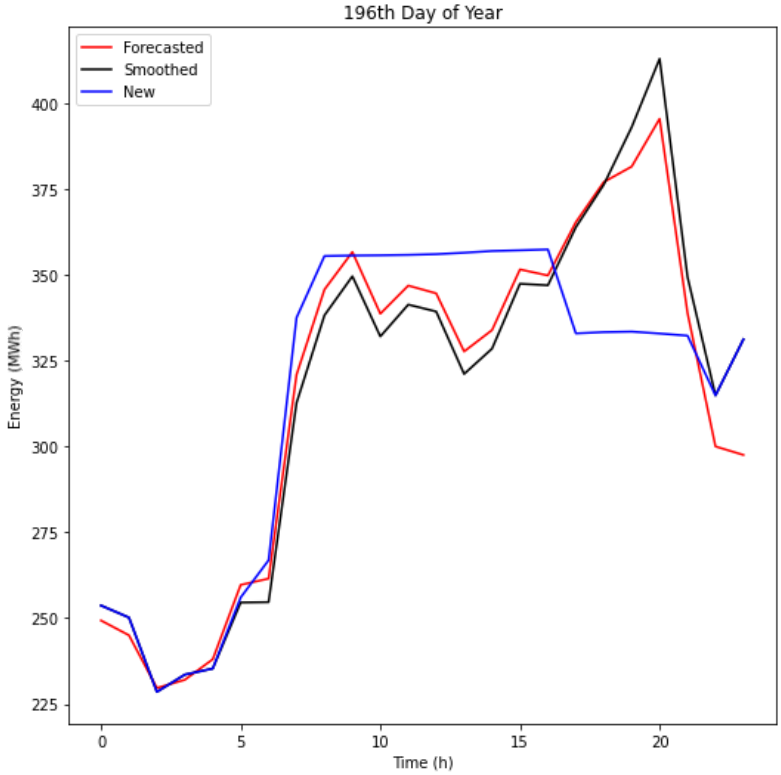


Figure 5.12 Output curve for day 196th

196th day comparison		
Forecasted (MWh)	Smoothed (MWh)	New (MWh)
420	380	350

Table 5.4 196th day peak energy comparison

From the given graph and table, it can be observed that in this case where there is not any need for extra energy during the next day, the energy on the phase 3 of the algorithm is less than the phase 1 and 2, respectively. It is observed that the total work produced is almost the same. The benefit of this process is the stabilization of the operation of the machine and therefore the reduction of consumption by working steadily for a long time. A more in-depth study could be done if the power plant performance graphs were available.

5.6 Model of the Islandic Power Plant

From the above actions emerges a final new model of energy production for the island. The original model consisted of the three forms of energy production and the island's energy needs, consumption. The new model has two electricity storage structures. The first was revised in section 5.3. The way to identify the needed battery capacity for the BESS is to find the charging and discharging areas in the year which is being studied. A reference week is depicted on the graph below.

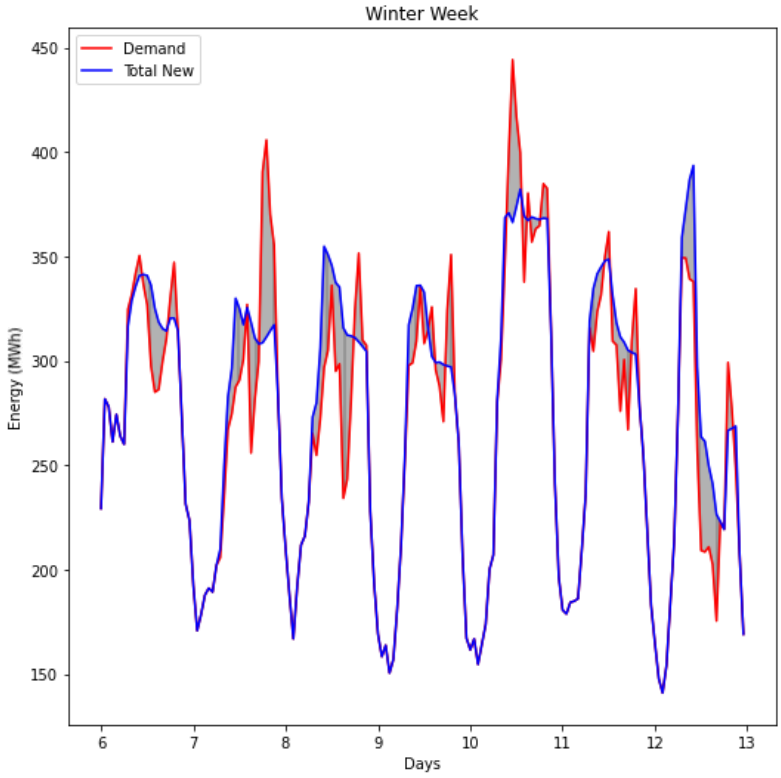


Figure 5.13 Winter Week algorithm curve

On the given figure, the blue and red line represent the New produced energy and the demand of energy respectively. The marked areas below the blue line and above the red line are the charging energy and the areas below the red and above the blue line are the discharging areas. They are determined by the following equations:

$$E_C = \int_0^T [P_n(t) - P_d(t)]dt$$

where $t \in [0, \ldots, T], P_{pr}\ and\ P_{sm}$ represent predicted and smoothed power, respectively, each time.

$$E_D = \int_0^T [\, P_d(t) \; - \; P_n(t)]dt$$

where $t \in [0, \ldots, T], P_{pr}\ and\ P_{sm}$ represent predicted and smoothed power, respectively, each time.

According the above, the needed capacity is 700MWh. Below it is depicted a figure of the Battery available power from the week used as an example above.

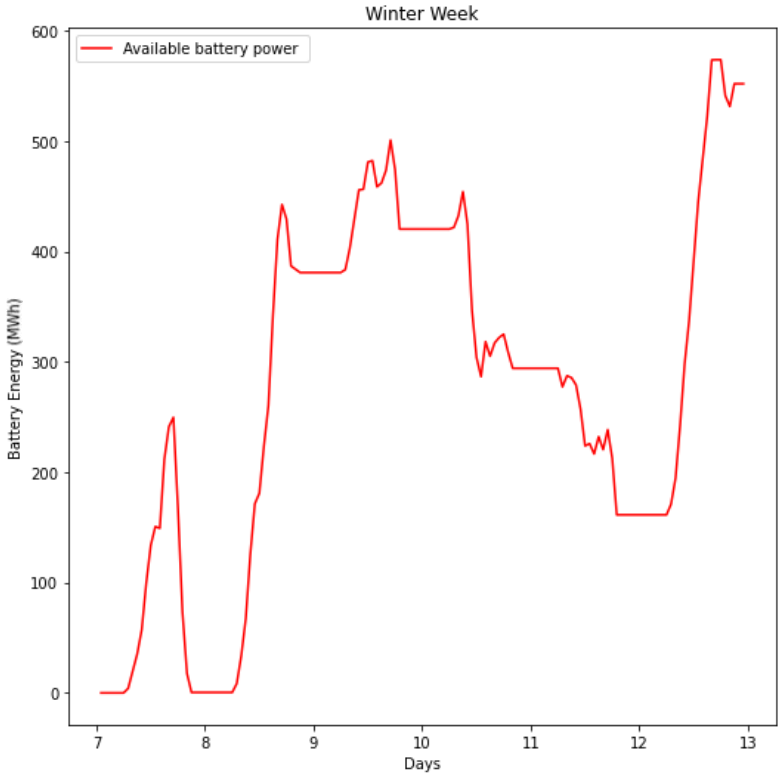


Figure 5.14 The available energy of the Battery of the winter week

6. Conclusions and Future Work

6.1 Conclusions

As society continually grows and expands the demand for power and energy consumption grows as well. With the technological advances being made, renewable energy is a significant part of the electric grid infrastructure. With any growth opportunity there can be positive and adverse effects. This thesis aims to provide a solution for mitigating electrical grid issues that arise with high levels of PV and Wind Energy penetration on a distribution feeder. In this, a predictive energy management system (EMS) based on load forecasting was introduced and integrated with the operation of a battery energy storage system (BESS). In this thesis, presented two different models of BESS. The first algorithm was developed to stabilize wind energy production. The factor that affects wind energy production is mainly due to the speed and direction of the wind, two factors that change frequently. It therefore makes sense to require the stabilization of the energy produced. on the other hand, solar energy is known to be produced during the day alone. in addition, it has much smaller instantaneous and marginal fluctuations in relation to wind energy, as it has and similar production in the same periods compared to the years. For this reason, a form of storage of extra solar energy that is not consumed was chosen to reduce the maximum load of energy produced by the thermal engine. In addition, in this diploma thesis the energy forecasting models are presented in detail.

The forecasting was done with deep learning methods. The neural networks were chosen to be built and trained in TensorFlow and Keras libraries in Python.

The results of the simulations showed that by applying the proposed methodology, it is possible to achieve a smoother operation of thermal machines as well as to replace them with lower power machines, while at the same time improving the utilization of the electricity generated by the photovoltaic park, allowing greater penetration during peak night hours.

6.2 Future Work

Considering the proposed simulation framework with the implementation of a predictive energy management system, developed in the framework of this thesis, there are many possible future steps that could improve the method and have a positive impact on the overall procedure.

First of all, a dynamic model it could be built. By creating a structural model the algorithm that was built in practice can be tested and its function can be proved. This could be further combined with a feasibility study to demonstrate the viability of the solution. Complete

energy production data such as fuel consumption and the specifications of all factors are necessary for this proof.

As a future work to be done is the amendment of this network into a Vehicle to Grid network. Vehicle-to-grid (V2G) is the latest attraction in field of EVs and their integration with electric grid. According to this phenomenon, bidirectional flow of electric power is taken into consideration, that is, power can be taken from grid to charge EV batteries during off peak hours and power can be provided to grid during peak hours from EV batteries to reduce utility load. A big portion of vehicles are expected to be parked during most part of the day. This idea can be used to facilitate V2G technology. During these idle times, plugged-in EVs can be used to support bidirectional power flow between utilities and EV batteries. These plugged-in EVs can provide ancillary services for utilities, such as, peak shaving, power quality improvement, and frequency and voltage regulation.

7. Tables

Table 3.1 System specification of hardware and software 26
Table 4.1 Correlation between inputs and PV energy 37
Table 4.2 Correlation between inputs and Wind energy 39
Table 4.3 Accuracy values comparison for three Neural Network Models 44
Table 4.4 Accuracy values comparison for three Neural Network Models 44
Table 4.5 Accuracy values comparison for three Neural Network Models 45
Table 4.6 Accuracy values comparison for three Neural Network Models 46
Table 5.1 Wind and Thermal total MWh before and after the smoothness for two weeks 55
Table 5.2 6th day peak energy comparison 61
Table 5.3 181st day peak energy comparison 62
Table 5.4 196th day peak energy comparison 62

8. Figures

Figure 1.1 Smart Grid Topology..... 9
Figure 2.1 Workflow of a machine learning problem. The data will be used to train a model with a machine learning algorithm. Then, in the prediction phase, the learned model can be used to generate a prediction. 12
Figure 2.2Shows typical curves for training and generalization error in dependency of the capacity of the model. Optimal capacity is reached at the minimal generalization error. Left of the optimal capacity is the model underfitting. On the right side of the optimal capacity is the model underfitting. On the right side of the optimal capacity is the model overfitting. The generalization error has typical a U-shaped curve. 13
Figure 2.3 Theses three diagrams show three different models, which tries to fit the sampled points. The sampled points are sampled from a noisy sinus function. The models are described by a polynom of degree {1,4,15} . The left model underfits the underlying function. The right model overfits the underlying function. It has the smallest error to fit the sampled points, but on unseen samples, it will provide a bad error. Idea for this figure is from [6]. 14
Figure 2.4 Feedforward Neural Network 16
Figure 2.5 Commonly known activation functions used in neural network models 17
Figure 2.6 An illustration of 4 standard recurrent units in a layer with their respective recurrent connections. The output signal is a function of the current input and recurrent input at time t and t-1, respectively. 21
Figure 2.7 Illustration of the convolution operation in CNNs for 2D-grids. The operation is performed for each filter on the input, resulting in an output volume with the same spatial 23
Figure 3.1 PV cell model 29
Figure 3.2 Temperature influence over the photovoltaic power generation [24] 30
Figure 3.3 PV innovation daily profile of the PV system installed in the examined island ... 30
Figure 4.1Yearly time period load curve correlation with temperature, for the test case system..... 32
Figure 4.2 Neural Network models 33
Figure 4.3 Developed network structure and data flow 34
Figure 4.4Correlation between PV Energy and Temperature..... 35
The solar irradiance and PV power forecasting methods are divided into physical and statistical models. The physical model mathematically or numerically manages the interaction of solar radiation in the atmosphere based on the laws of physics. It comprises numerical weather prediction, sky imagery, and satellite image models. The statistical model finds a relationship between input and output variables and consists of conventional statistical models and machine learning models. Conventional statistical models include the fuzzy theory, Markov chain, autoregressive, and regression models. The machine learning

model, also known as an artificial intelligence model, can efficiently extract high dimensional complex nonlinear features and directly map input and output variables. In the past, the well-known machine learning models for predicting solar energy were the support vector machine (SVM), k-nearest neighbors, artificial neural network (ANN), naïve Bayes, and Figure 4.5 Correlation between PV Energy and Solar irradiance 35
Figure 4.6 Developed network structure and data flow 38
Figure 4.7 Correlation between Wind Energy and Wind speed 39
Figure 4.8 Wind speed before and after the application of Kalman filter 40
Figure 4.9 Developed network structure and data flow 41
Figure 4.10 Graphic Correlation between Thermal Energy and Temperature 41
Figure 4.11 Graphic Correlation between Thermal Energy and Demand 42
Figure 4.12 Developed network structure and data flow 43
Figure 4.13 Results Compared for Total Demand 44
Figure 4.14 Results Compared for Solar Energy 45
Figure 4.15 Results Compared For Wind Energy 46
Figure 4.16 Results Compared for Thermal Energy 47
Figure 5.1 Demand curve in the first 10 days of the Year 2016 48
Figure 5.2 The “duck” net load curve, illustrating steep ramp needs and overgeneration risk [38] 50
Figure 5.3 Wind predicted energy curve 50
Figure 5.4 Wind BESS simulation 51
Figure 5.5 Correlation between Predicted and Smoothed power curve 53
Figure 5.6 Operation of peak shaving algorithm in a random day 56
Figure 5.7 Comparison of Final Curve with Demand 57
Figure 5.8 Artificially created excess energy and the corresponding demand, for day 6 58
Figure 5.9 Algorithm output curve for 6th day 59
Figure 5.10 Algorithm output curve for 181st day 60
Figure 5.11 Comparison of two curves 61
Figure 5.12 Output curve for day 196th 62
Figure 5.13 Winter Week algorithm curve 63
Figure 5.14 The available energy of the Battery of the winter week 64

9. List of References

[1] *Dolf Gielen, Francisco Boshell, Deger Saygin, Morgan D. Bazilian, Nicholas Wagner, Ricardo Gorini, The role of renewable energy in the global energy transformation, Energy Strategy Reviews, Volume 24, 2019, Pages 38-50, ISSN 2211-467X, https://doi.org/10.1016/j.esr.2019.01.006*

[2] *C. Cecati, G. Mokryani, A. Piccolo and P. Siano, "An overview on the smart grid concept," IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society, Glendale, AZ, USA, 2010, pp. 3322-3327, doi: 10.1109/IECON.2010.5675310*

[3] *Spyridon Chapaloglou, Athanasios Nesiadis, Petros Iliadis, Konstantinos Atsonios, Nikos Nikolopoulos, Panagiotis Grammelis, Christos Yiakopoulos, Ioannis Antoniadis, Emmanuel Kakaras, Smart energy management algorithm for load smoothing and peak shaving based on load forecasting of an island's power system, Applied Energy, Volume 238, 2019, Pages 627-642, ISSN 0306 2619, https://doi.org/10.1016/j.apenergy.2019.01.102*

[4] *Thomas M. Mitchell. Machine Learning. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997. ISBN 9780070428072.*

[5] *Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani, eds. An introduction to statistical learning: with applications in R. Springer texts in statistics 103. OCLC: ocn828488009. New York: Springer, 2013. 426 pp. ISBN: 978-1-4614- 7137-0*

[6] *Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep Learning. Book in preparation for MIT Press, 2016. URL http://www.deeplearningbook.org*

[7] *Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 1097–1105. Curran Associates, Inc., 2012*

[8] *Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. CoRR, abs/1512.00567, 2015. URL http://arxiv.org/abs/1512.00567*

[9] *Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In David Blei and Francis Bach, editors, Proceedings of the 32nd International Conference on Machine Learning (ICML-15), volume 37, pages 448–456. JMLR Workshop and Conference Proceedings, 2015*

[10] *Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout : A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research (JMLR), 15:1929–1958, 2014*

[11] *Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann Lecun, and Christoph Bregler. Efficient object localization using Convolutional Networks. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 648–656, June 2015*

[12] *Sepp Hochreiter and Jürgen Schmidhuber. "LONG SHORT-TERM MEMORY". In: MIT Press Cambridge, MA, USA (Nov. 15, 1997). URL: http://www.bioinf.jku. at/publications/older/2604.pdf*

[13] *Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. "WaveNet: A Generative Model for Raw Audio". In: arXiv:1609.03499 [cs] (Sept. 12, 2016). arXiv: 1609.03499. URL: http://arxiv.org/ abs/1609.03499*

[14] *Shaojie Bai and J. Zico Kolter and Vladlen Koltun, An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling, arXiv:1803.01271,2018*

[15] *G. Box, G. Jenkins, Time Series Analysis: Forecasting and Control, San Francisco: Holden-Day, 1970*

[16] *A. Earnest, M. I. Chen, D. Ng, L. Y. Sin, "Using Autoregressive Integrated Moving Average (ARIMA) Models to Predict and*

[17] *Monitor the Number of Beds Occupied During a SARS Outbreak in a Tertiary Hospital in Singapore," in BMC Health Service Research, 5(36), 2005*

[18] *M. J. Kane, N. Price, M. Scotch, P. Rabinowitz, "Comparison of ARIMA and Random Forest Time Series Models for Prediction of Avian Influenza H5N1 Outbreaks," BMC Bioinformatics, 15(1), 2014*

[19] *Python - Programming language. Python.org. URL: https : / / www . python . org/*

[20] *A. Earnest, M. I. Chen, D. Ng, L. Y. Sin, "Using Autoregressive Integrated Moving Average (ARIMA) Models to Predict and*

[21] *Wes McKinney. "Pandas: a Foundational Python Library for Data Analysis and Statistics". In: Python High Performance Science Computer (Jan. 2011), p. 9. URL: https : / / www . dlr . de / sc / Portaldata / 15 / Resources / dokumente / pyhpc2011 /*

submissions/pyhpc2011_submission_9.pdf.] and Keras [François Chollet. Keras - Neural network library. Keras: The Python Deep Learning library. URL: https://keras.io/

[22] *Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. "TensorFlow: A system for large-scale machine learning". In: arXiv:1605.08695 [cs] (May 27, 2016). arXiv: 1605.08695. URL: http://arxiv.org/abs/1605.08695*

[23] *John Nickolls, Ian Buck, Michael Garland, and Kevin Skadron. "Scalable Parallel Programming with CUDA". In: Queue 6.2 (Mar. 2008), pp. 40–53. ISSN: 1542-7730. DOI: 10.1145/1365490.1365500. URL: http://doi.acm.org/10.1145/ 1365490.1365500*

[24] *Hersch, P, and Zweibel, K. Basic photovoltaic principles and methods. United States: N. p., 1982. Web. doi:10.2172/5191389*

[25] *M. Bilgin, Analyzing the effect of surface temperature on the efficiency of photovoltaic panels, [M.S. Thesis], Marmara University, İstanbul, Turkey, 2013*

[26] *Kim Y, Hur J. An Ensemble Forecasting Model of Wind Power Outputs Based on Improved Statistical Approaches.* Energies*. 2020; 13(5):1071. https://doi.org/10.3390/en13051071*

[27] *Das, U.K.; Tey, K.S.; Seyedmahmoudian, M.; Mekhilef, S.; Idris, M.Y.I.; Van Deventer, W.; Horan, B.; Stojcevski, A. Forecasting of photovoltaic power generation and model optimization: Areview. Renew. Sustain. Energy Rev. 2018, 81, 912–928*

[28] *Ng, A. Machine Learning Yearning: Techincal Strategy for AI Engineers, in the Era of Deep Learning. 2018. Available online: https://www.deeplearning.ai/machine-learning-yearning*

[29] *Zang, H.; Liu, L.; Sun, L.; Cheng, L.;Wei, Z.; Sun, G. Short-term global horizontal irradiance forecasting based on a hybrid CNN-LSTM model with spatiotemporal correlations. Renew. Energy 2020, 160, 26–41*

[30] *Zhao, P.; Wang, J.; Xia, J.; Dai, Y.; Sheng, Y.; Yue, J. Performance evaluation and accuracy enhancement of a day-ahead wind power forecasting system in China. Renew. Energy 2012, 43, 234–241*

[31] *Kim, T.Y.; Cho, S.B. Predicting residential energy consumption using CNN-LSTM neural networks. Energy 2019, 182, 72–81*.

[32] *H. C. Sung, J. B. Park, and Y. H. Joo, "Robust observer-based fuzzy control for variable speed wind power system : LMI approach", Int. Journal of Control, Automation, and Systems, vol. 9, no. 6, pp. 1103- 1110, 2011, 12*

[33] *A. Uehara, A. Pratap, T. Goya and T. Senjyu, "A coordinated control method to smooth wind power fluctuations of a PMSG-based WECS," Energy Conversion, vol. 26, no. 2, pp. 550-558, 2011, 6*

[34] *Mohammad Taghi Zareifard. Modelling and Control of Wind Farms Integrated with Battery Energy Storage Systems. [Research Report] university of new south wales 2017. ⟨hal-01573257⟩*

[35] *E Alessio, A Carbone, G Castelli, and V Frappietro. "Second-order moving average and scaling of stochastic time series". In: The European Physical Journal B-Condensed Matter and Complex Systems 27.2 (2002), pages 197–200. doi: 10.1140/epjb/ e20020150*

[36] *E Alessio, A Carbone, G Castelli, and V Frappietro. "Second-order moving average and scaling of stochastic time series". In: The European Physical Journal B-Condensed Matter and Complex Systems 27.2 (2002), pages 197–200. doi: 10.1140/epjb/ e20020150*

[37] *S. X. Chen, H. B. Gooi and M. Q. Wang, "Sizing of Energy Storage for Microgrids," in* IEEE Transactions on Smart Grid, *vol. 3, no. 1, pp. 142-151, March 2012, doi: 10.1109/TSG.2011.2160745*

[38] *California Independent System Operator, What the duck curve tells us about managing a green grid, http://large.stanford.edu/courses/2015/ph240/burnett2/docs/flexible.pdf*

10. Greek Extended Summary

10.1 Εισαγωγή – Περιγραφή του προβλήματος

Στην παρούσα διπλωματική εργασία μελετήθηκαν δύο θέματα. Αρχικά μελετήθηκαν και συγκρίθηκαν μέθοδοι πρόβλεψης χρονοσειρών οι οποίες βασίζονταν σε μεθόδους βαθιάς μάθησης (Deep Learning). Έπειτα, μελετήθηκε η ενσωμάτωση «έξυπνων» αλγορίθμων στο σύστημα διαχείρισης ηλεκτρικής ενέργειας ενός νησιού. Η διπλωματική αυτή βασίστηκε σε πραγματικά δεδομένα κατανάλωσης και παραγωγής ενέργειας από το έτος 2014 έως και το 2016. Οι υλοποιήσεις των αλγορίθμων βασίστηκαν σε μοντέλα αποθήκευσης ηλεκτρικής ενέργειας (BESS). Οι πηγές παραγωγής ενέργειας ήταν η συμβατική μέθοδος παραγωγής μέσω θερμικής μηχανής (Diesel και LNG), αιολική και ηλιακή ενέργεια. Αφού έγινε η πρόβλεψη των χρονοσειρών ακολούθησε ο σχεδιασμός των δύο αλγορίθμων. Ο πρώτος αλγόριθμος αφορούσε την διαχείριση της αιολικής ενέργειας και την εξομάλυνση αυτής και κατ' επέκταση την εξομάλυνση και τη σταθεροποίηση του παραγόμενου φορτίου από τη θερμική μηχανή. Ο δεύτερος αλγόριθμος αφορούσε το peak shaving με μεθόδους αποθήκευσης ενέργειας (BESS). Το σύστημα αποθηκεύει την ενέργεια στην BESS κάθε φορά που υπάρχει πλεόνασμα στην παραγωγή από το φωτοβολταϊκό σύστημα και το χρησιμοποιεί ως επιπλέον πηγή ενέργειας όταν χρειάζεται, για παράδειγμα κατά τη διάρκεια της νύχτας. Στην περίπτωση της Αιολικής Ενέργειας, μέσω ενός αλγορίθμου βελτιστοποίησης εντοπίζεται η βέλτιστη καμπύλη και η BESS λειτουργεί ως σταθεροποιητής παραγωγής σε αυτή την καμπύλη. Μια πρόκληση για την εξεύρεση λύσης έγκειται στην έλλειψη δεδομένων παραγωγής μεγάλων φωτοβολταϊκών και αιολικών σταθμών που είναι σε θέση να δημιουργήσουν διαταράξεις σε μικρότερα δίκτυα. Αυτά τα δεδομένα είναι απαραίτητα για την πρόβλεψη πιθανών μελλοντικών συμπεριφορών του εργοστασίου και τον έλεγχο της τάσης του συστήματος και των επιπέδων ισχύος (τόσο ενεργών όσο και αντιδραστικών) για την αποφυγή διαταραχών του δικτύου. Η ανάπτυξη αυτής της ιδέας βασίζεται στα Smart Grids. Το Smart Grid είναι μια έννοια για τη μετατροπή του δικτύου ηλεκτρικής ενέργειας χρησιμοποιώντας προηγμένες τεχνικές αυτόματου ελέγχου και επικοινωνιών και άλλες μορφές τεχνολογίας πληροφοριών. Ενσωματώνει καινοτόμα εργαλεία και τεχνολογίες από την παραγωγή, τη μεταφορά και τη διανομή μέχρι τις καταναλωτικές συσκευές και εξοπλισμό. Η έννοια αυτή ενσωματώνει τις ενεργειακές υποδομές, τις διαδικασίες, τις συσκευές, τις πληροφορίες και τις αγορές σε μια συντονισμένη και συνεργατική διαδικασία που επιτρέπει την παραγωγή, τη διανομή και την αποδοτικότερη κατανάλωση ενέργειας [2]. Πριν το σχεδιασμό του αλγορίθμου πραγματοποιήθηκε η πρόβλεψη χρονοσειρών. Οι προβλεπόμενες παράμετροι ήταν η συνολική ζήτηση φορτίου και οι παραγωγές θερμικής, ηλιακής και αιολικής ενέργειας. Η πρόβλεψη έγινε με αλγόριθμους νευρωνικού δικτύου που συνδυάζουν συνελικτικά και επαναλαμβανόμενα νευρωνικά δίκτυα (RNN και CNN αντίστοιχα). Σε σύγκριση με άλλες μελέτες, η παρούσα μελέτη διαφέρει καθώς μελετά την πρόβλεψη όλων των πηγών ενέργειας, τη διαχείριση αυτών των προβλεπόμενων τιμών μέσω ενός ευφυούς

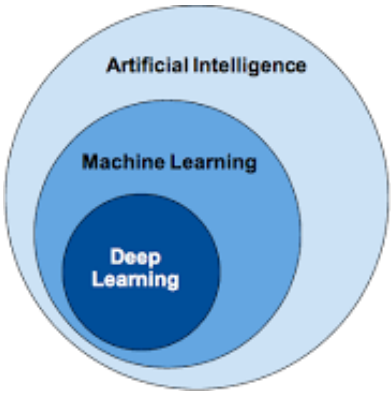
αλγορίθμου που χρησιμοποιεί μηχανική μάθηση και στη συνέχεια δοκιμάζει τη λειτουργία της σε ένα μοντέλο.

11. Ανάλυση Μεθόδου

11.1 Θεωρία Βαθιάς Μάθησης

Η Βαθιά Μάθηση είναι ένα υποπεδίο της μηχανικής μάθησης, η οποία είναι υποπεδίο του τομέα της Τεχητητής Νοημοσύνης.

Η κεντρική ιδέα της βαθιάς μάθησης είναι εμπνευσμένη από τη βιολογική συμπεριφορά του εγκεφάλου. Με απλά λόγια, κάθε νευρώνας, ο δομικός λίθος του εγκεφάλου, μεταδίδει πληροφορία σε γειτονικού νευρώνες, σχηματίζοντας ένα μεγάλα και πολύπλοκο δίκτυο. Κάθε κόμβος ή νευρώνας διεγείρεται απο τις εισροές και μεταδίδει την πληροφορία ή κομμάτι αυτής σε άλλους κόμβους.



Σχήμα 1: Τα πεδία Τεχνητή Νοημοσύνη, Μηχανική Μάθηση και Βαθιά Μάθηση

11.1.1 Επιτηρούμενη και Μη-Επιτηρούμενη Μάθηση

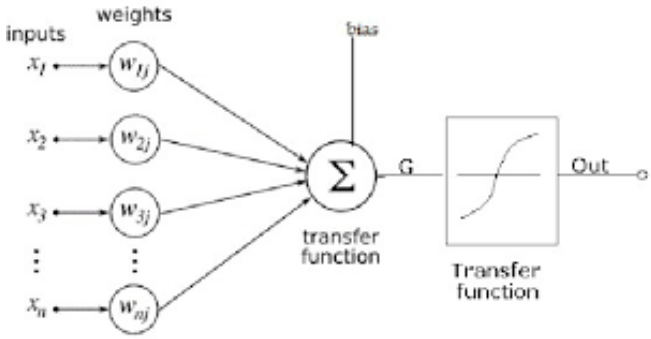
Στην μηχανική μάθηση υπάρχουν δύο ομάδες αλγορίθμων που διαφέρουν μεταξύ τους στον τρόπο με τον οποίο μαθαίνουν, η Επιτηρούμενη και η Μη-Επιτηρούμενη μάθηση.

Στην Επιτηρούμενη μάθηση έχουμε δείγματα από επιγραφόμενα δεδομένα και ο αλγόριθμος μαθαίνει να προβλέπει το αποτέλεσμα βασισμένος στα εισαγώμενα δεδομένα. Ποιο αυστηρά, δωθέντος ενός σετ N παραδειγμάτων εκπαίδευσης της μορφής (x_1, y_1), …, (x_N, y_N) τέτοια ώστε το x_i είναι το χαρακτηριστικό διάνυσμα του i-οστου παραδείγματος και y_i η επιγραφή του, ένα αλγόριθμος μάθησης αναζητά μία συνάρτηση $f : X \rightarrow Y$, όπου X ο χώρος των εισαγώμενων δεδομένων και Y ο χώρος των αποτελεσμάτων. Η παρούσα διπλωματική εργασία βασίζεται σε αυτή τη μέθοδο.

Στην Μη-Επιτηρούμενη Μάθηση από την άλλη δεν χρειζόμαστε επιγραφόμενα δεδομένα. Ο αλγόριθμος είναι αυτός που προσπαθεί να βρεί δομή στα δεδομένα. Μπορεί να χρησιμοποιηθεί ώστε να κατηγοριοποιήσει μη επιγραφόμενα δεδομένα.

11.1.2 Τεχνητός Νευρώνας

Ο τεχνητός νευρώνας είναι το δομικό στοιχείο του νευρωνικού δικτύου. Αποτελείται από εισροές και εκροές, βάρη κλίση και μια συνάρτηση ενεργοποίησης ή μεταφοράς.



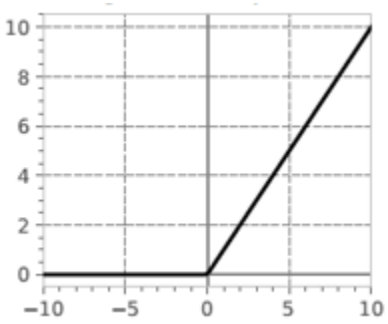
Σχήμα 2: Τεχνητός Νευρώνας

11.1.3 Συνάρτηση Ενεργοποίησης

Διαφορετικές συναρτήσεις ενεργοποίησης χρησιμοποιούνται για διαφορετικά προβλήματα. Στο παρόν πρόβλημα επιλέχθηκε να χρησιμοποιηθεί η Διορθωμένη Γραμμική, γνωστή και ως ReLU Activation Function. Η Διορθωμένη Γραμμική είναι η πιο συνηθισμένη συνάρτηση ενεργοποίησης λόγω της απλότητάς της και των καλών αποτελεσμάτων της. Ένα υποσύνολο νευρώνων ενεργοποιείται κάθε φορά. Αυτό κάνει το δίκτυο πιο αραιό, βελτιώνοντας την απόδοσή του. Με μια ομοιόμορφη αρχικοποίηση των βαρών, περίπου οι μισοί από τους κρυμμένους νευρώνες θα ενεργοποιηθούν. Ο τύπος αυτής της λειτουργίας είναι απλός:

$$\sigma(z) = \max(0, z) = \begin{cases} 0, & if\ z < 0 \\ z, & if\ z \geq 0 \end{cases}$$

Η Διορθωμένη γραμμική απεικονίζεται παρακάτω.

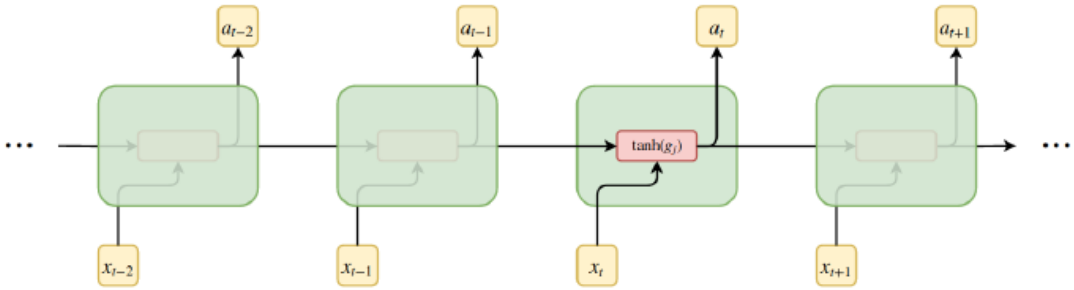


Σχήμα 3: Διορθωμένη Γραμμική Συνάρτηση Ενεργοποίησης

11.1.4 Μοντέλα Νευρωνικών Δικτύων

11.1.4.1 Recurrent Neural Networks

Ένα απλό νευρωνικό δίκτυο, αλλά όπου οι κόμβοι σε κάθε επίπεδο έχουν τώρα ενδιάμεσες συνδέσεις (επαναλαμβανόμενες συνδέσεις). Το αποτέλεσμα είναι ένα επαναλαμβανόμενο νευρωνικό δίκτυο (RNN) όπως φαίνεται στο Σχήμα παρακάτω, όπου οι συνδέσεις αντιπροσωπεύουν χρονικές εξαρτήσεις (χρονικές εξαρτήσεις) και εισάγουν ένα επιπλέον σύνολο συντελεστών στάθμισης με δυνατότητα βελτιστοποίησης. Τα RNNs είναι μια άλλη οικογένεια νευρωνικών δικτύων που έχουν σχεδιαστεί για εφαρμογές σε διαδοχικά προβλήματα όπως η επεξεργασία γλωσσών και τα πιο απλά RNN εισάγουν μόνο αυτές τις επαναλαμβανόμενες συνδέσεις.



Σχήμα 4: Μια απεικόνιση 4 τυπικών RNN σε ένα επίπεδο με τις αντίστοιχες επαναλαμβανόμενες συνδέσεις τους. Το σήμα εξόδου είναι συνάρτηση της τρέχουσας εισόδου και της περιοδικής εισόδου τη χρονική στιγμή *t* και *t-1*, αντίστοιχα.

11.1.4.2 Convolutional Neural Networks

Έστω ένα τυπικό νευρωνικό δίκτυο τροφοδοσίας. Υπάρχουν δύο ανησυχίες κατά την εφαρμογή ενός Feed-Forward neural network σε τοπολογίες που μοιάζουν με πλέγμα με περισσότερες από μία διαστάσεις. Το πρώτο είναι ότι οι χωρικές πληροφορίες δεν διατηρούνται, καθώς το δίκτυο πρέπει να ισοπεδωθεί στα δίκτυα. Δεύτερον, μια εφαρμογή όπως αυτή δεν κλιμακώνεται για περιπτώσεις χρήσης όπως η ταξινόμηση εικόνων και άλλες τοπολογίες που μοιάζουν με πλέγμα, λόγω του αυξημένου αριθμού παραμέτρων ως αποτέλεσμα πολλών διασυνδεδεμένων νευρώνων στο πλέγμα. Μια τέτοια αύξηση θα απαιτούσε περισσότερους υπολογιστικούς πόρους. Τα CNN επιλύουν αυτά τα προβλήματα εξετάζοντας μόνο ένα υποσύνολο της εισόδου, γνωστό ως τοπική συνδεσιμότητα.

12. Γενικές Πληροφορίες

12.1 Δεδομένα

Η βάση δεδομένων στην οποία πραγματοποιήθηκε η διαδικασία εκμάθησης αποτελείται από ωριαίες τιμές παραγωγής και κατανάλωσης ενέργειας για τρία συναπτά έτη σε ένα νησί. Επιπλέον, υπάρχουν δεδομένα κλιματολογικών συνθηκών, όπως θερμοκρασία αέρα και ταχύτητα ανέμου. Σε πρώτη φάση ανάλυσης των δεδομένων βρέθηκε ο βαθμός συσχέτισης που έχουν οι διάφορες κλιματολογικές συνθήκες με την παραγωγή και την κατανάλωση ενέργειας. Με αυτό τον τρόπο επιλέχθηκαν οι κατάλληλες τιμές εισόδου για τις αντίστοιχες τιμές εξόδου.

12.2 Λεπτομέρειες Συστήματος

The Python programming language [19] is used to perform data analysis, develop models, run experiments and evaluations. With a rich ecosystem and a diverse set of supported libraries and frameworks, tasks related to data analysis and neural network modelling are more convenient. Further on, Table 2.1 summarises what software, hardware and frameworks that are used.

Η γλώσσα προγραμματισμού Python [19] χρησιμοποιήθηκε για την ανάλυση δεδομένων, την ανάπτυξη μοντέλων, την εκτέλεση πειραμάτων και αξιολογήσεων. Με ένα πλούσιο σύνολο υποστηριζόμενων βιβλιοθηκών και πλαισίων, οι εργασίες που σχετίζονται με την ανάλυση δεδομένων και τη μοντελοποίηση νευρωνικών δικτύων μπορούν να παρέχουν μεγαλύτερη ακρίβεια και εξατομίκευση. Επιπλέον, ο πίνακας 3.1 συνοψίζει το λογισμικό, το υλικό και τα πλαίσια που χρησιμοποιούνται.

Software		
Name	**Version**	**Description**
MS Windows 10 Home	10.0.18363	Operating System
Python		Used for implementation
Keras		Used for nuilding models
Pandas		Used for data analysis
TensorFlow		Used as backend for Keras
Cuda		Required for Tensorflow
Hardware		
Name		**Description**
CPU		Intel i5-8300H
GPU		NVIDIA GTX 1050Ti
Memory		8.00GB
GPU Memory		4.00GB

Πίνακας 1: Λεπτομέρειες Συστήματος

13. Διαδικασία Πρόβλεψης

Ένα μοντέλο πρόβλεψης είναι ένα απαραίτητο υποσύστημα που πρέπει να εφαρμοστεί σε έναν αλγόριθμο προληπτικής διαχείρισης ενέργειας ικανό να αντισταθμίσει μελλοντικά συμβάντα. Η ανάγκη πρόβλεψης αφορά την κατανάλωση και τις παραγωγές ενέργειας(Θερμική, Αιολική και Ηλιακή ενέργεια). Πρώτο βήμα για την έναρξη της προβλεπτικής διαδικασίας είναι η επιλογή τιμών εισόδου για κάθε επιλεγμένη έξοδο. Η επιλογή βασίζεται τόσο σε δεδομένα βιβλιογραφίας όσο και στο δείκτη συσχέτισης μεταξύ εισόδου και εξόδου. Για κάθε επιλεγμένη έξοδο κατασκευάστηκαν τρία μοντέλα πρόβλεψης, ένα LSTM, ένα CNN και ένα Υβριδικό που αποτελείται από CNN και LSTM.

Πριν από κάθε διαδικασία εκμάθησης, εφαρμόστηκε η συνάρτηση MinMax Scaler ώστε όλα να γίνουν συμπτυχθούν μεταξύ του [0,1].

$$MinMax\ Scaler = \frac{x_i - mean(x)}{\max(x) - \min(x)}$$

Επιπλέον για να στηθεί σωστά ένας αλγόριθμος πρόβλεψης πρέπει να γίνει ένας διαχωρισμός μεταξύ δεδομένων εκμάθησης και δοκιμής (training-test split). Για να επιτευχθεί αυτό ολοκληρωμένα πρέπει να ισχύουν τα εξής δύο: Καταρχάς, πρέπει να είναι αρκετά μεγάλο το μέγεθος των δεδομένων δοκιμής για να αποφέρει στατιστικά σημαντικά αποτελέσματα. Δεύτερον, το σύνολο δοκιμών πρέπει να είναι αντιπροσωπευτικό του συνόλου δεδομένων στο σύνολό του. Με άλλα λόγια, αυτό το σύνολο δεδομένων δεν πρέπει να περιέχει διαφορετικά χαρακτηριστικά από το σύνολο εκπαίδευσης. Γενικά, το μέγεθος του δείγματος σχετίζεται στενά με την απαιτούμενη ακρίβεια του προβλήματος. Όσο μεγαλύτερο είναι το μέγεθος, τόσο πιο ακριβή θα είναι τα αποτελέσματα. Για το συγκεκριμένο πρόβλημα, τα μοντέλα και οι επιλεγμένοι υπερπαράμετροι (hyperparameters) δικτύου δεν αντιπροσώπευαν υψηλή πολυπλοκότητα, επομένως, το μέγεθος των δεδομένων δεν ήταν περιοριστικός παράγοντας της ακρίβειας που επιτεύχθηκε. Επειδή το σύνολο δεδομένων ήταν αρκετά μεγάλο, ήταν εφικτό να υπάρχει διαχωρισμός 70/30, 70% του συνολικού συνόλου δεδομένων που χρησιμοποιήθηκε για την εκπαίδευση και 30% για τις δοκιμές. Όπως είναι λογικό για κάθε διαφορετική παράμετρο που επρόκειτο να ελεγχθεί επιλέχθηκαν διαφορετικές παράμετροι εισόδου. Έτσι προέκυψαν τα εξής:

- Παράμετροι εισόδου και εξόδου Ολικού Φορτίου:
    - 48 τιμές των δεδομένων ωριαίας κατανάλωσης των δύο προηγούμενων ημερών,
    - 24 τιμές δεδομένων θερμοκρασίας προηγούμενης ημέρας
    - 7 δυαδικές τιμές που αντιστοιχούν στην ημέρα της εβδομάδας

- Η έξοδος αποτελούνταν από ένα διάνυσμα 24 μεταβλητών που περιείχε τις προβλεπόμενες τιμές φορτίου της επόμενης ημέρας

- Παράμετροι εισόδου και εξόδου Ηλιακής Ενέργειας:
  - 7 δυαδικές τιμές που αντιστοιχούν στην ημέρα της εβδομάδας
  - 24 τιμές δεδομένων θερμοκρασίας προηγούμενης ημέρας
  - 24 τιμές προηγούμενων διάχυτων δεδομένων ακτινοβολίας
  - 24 τιμές προηγούμενων δεδομένων άμεσης ακτινοβολίας
  - Η έξοδος αποτελούνταν από ένα διάνυσμα 24 μεταβλητών που περιείχε τις προβλεπόμενες τιμές παραγωγής Ηλιακής Ενέργειας της επόμενης ημέρας
- Παράμετροι εισόδου και εξόδου Αιολικής Ενέργειας:
  - 7 δυαδικές τιμές που αντιστοιχούν στην ημέρα της εβδομάδας
  - 24 τιμές προηγούμενων δεδομένων ταχύτητας ανέμου
  - Η έξοδος αποτελούνταν από ένα διάνυσμα 24 μεταβλητών που περιείχε τις προβλεπόμενες τιμές παραγωγής Αιολικής Ενέργειας της επόμενης ημέρας
- Παράμετροι εισόδου και εξόδου Θερμικής Ενέργειας:
  - 7 δυαδικές τιμές που αντιστοιχούν στην ημέρα της εβδομάδας
  - 24 τιμές των δεδομένων ωριαίας κατανάλωσης της προηγούμενης ημέρας,
  - 24 τιμές δεδομένων θερμοκρασίας προηγούμενης ημέρας
  - Η έξοδος αποτελούνταν από ένα διάνυσμα 24 μεταβλητών που περιείχε τις προβλεπόμενες τιμές παραγωγής Θερμικής Ενέργειας της επόμενης ημέρας

Από αυτά το μοντέλα προέκυψαν κάποιες τιμές εξόδου που συγκρίθηκαν με τις πραγματικές τιμές. Έτσι προέκυψαν κάποιοι βαθμοί ακρίβειας για το κάθε μοντέλο. Οι μετρήσεις αξιολόγησης των επιδόσεων για κάθε μοντέλο βασίστηκαν στα εξής:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2}$$

$$MAE = \frac{\sum_{i=1}^{n}(Y_i - \hat{Y}_i)}{n}$$

Συγκρίνοντας τα εκάστοτε σφάλματα προέκυψαν τα ακόλουθα τελικά μοντέλα πρόβλεψης:

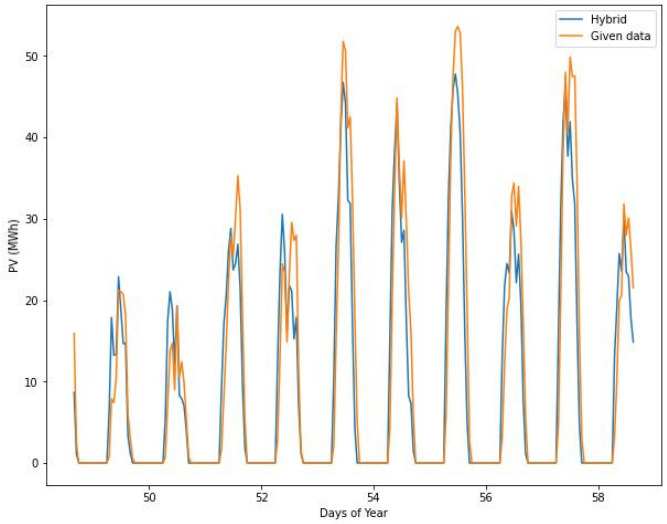
Total Forecast	**MAE**	**RMSE**	**MSE**
Hybrid	0.054	0.069	0.005

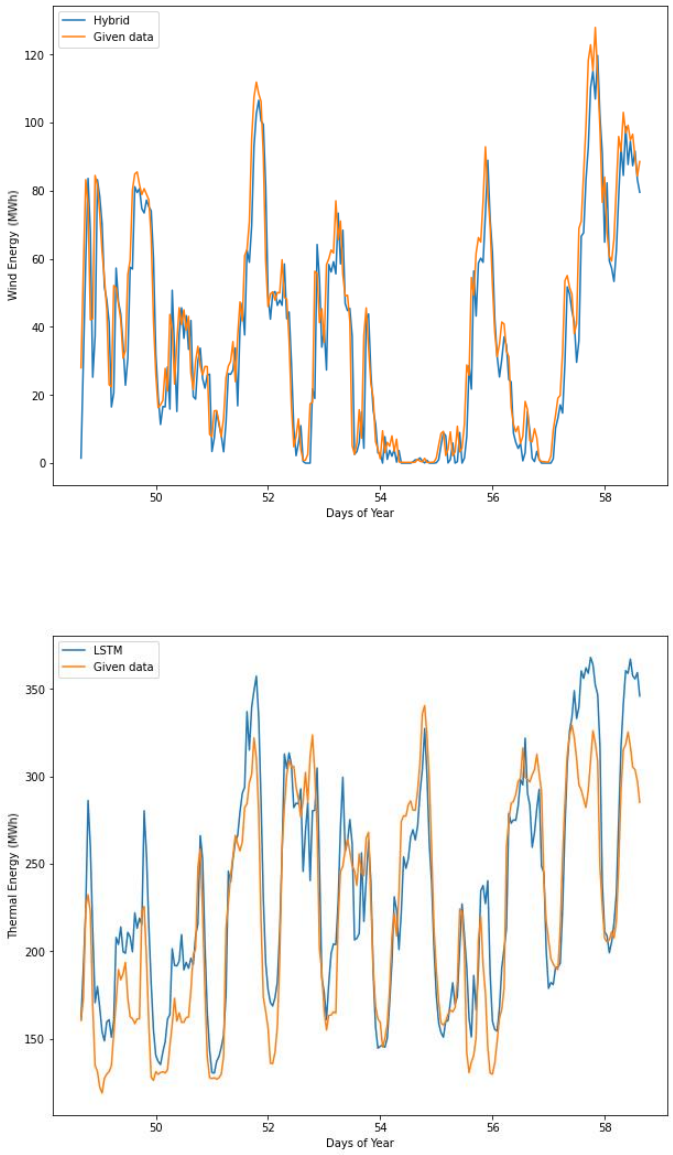
Solar	**MAE**	**RMSE**	**MSE**
Hybrid	0.039	0.068	0.005

Wind	**MAE**	**RMSE**	**MSE**
Hybrid	0.048	0.065	0.004

Thermal	**MAE**	**RMSE**	**MSE**
LSTM	0.070	0.068	0.005

Πίνακας 2: Τα μοντέλα που προέκυψαν με τις χαμηλότερες τιμές σφαλμάτων





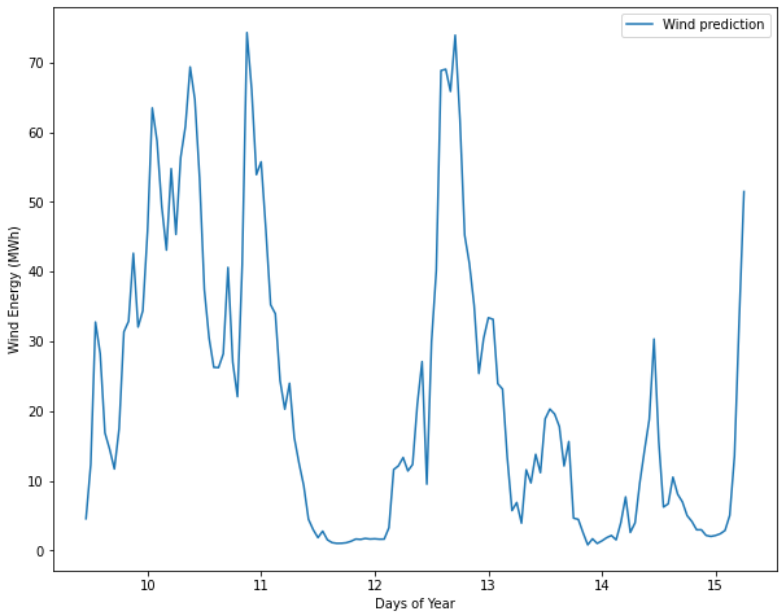
Σχήμα 5: Αποτυπώνεται σύγκριση μεταξύ των δεδομένων που προέκυψαν από τα τελικά μοντέλα προσομοίωσης και των αληθινών δεδομένων

14. Αλγόριθμος Βελτιστοποίησης

Αναπτύχθηκε ένας αλγόριθμος για να ενσωματωθεί στο σύστημα διαχείρισης ενέργειας του δικτύου νησιωτικής ενέργειας, ο οποίος θα μπορούσε να αναβαθμίσει τη λειτουργία του παρέχοντας ανανεώσιμη ενέργεια με πιο ασφαλή και σταθερό τρόπο και με μέγιστο ξύρισμα της μέγιστης τιμής ζήτησης. Ο αναπτυγμένος αλγόριθμος ήταν κατάλληλος για τη συγκεκριμένη νησιωτική περίπτωση που ερευνήθηκε σε αυτή τη μελέτη, αλλά θα μπορούσε επίσης να εφαρμοστεί με λίγες τροποποιήσεις σε άλλα παρόμοια συστήματα ισχύος. Μετά τη διαμόρφωση και τη δοκιμή της μονάδας πρόβλεψης φορτίου, είναι δυνατόν να προβλεφθούν οι τιμές ζήτησης και παροχής της επόμενης ημέρας σε ωριαία βάση. Αυτή η ικανότητα ελήφθη υπόψη για την ανάπτυξη ενός προγνωστικού αλγορίθμου συστήματος διαχείρισης ενέργειας (EMS) που θα μπορούσε να χρησιμοποιηθεί για την «εξυπνοποίηση» του συστήματος ηλεκτρικής ενέργειας.

14.1 Αλγόριθμος Αιολικής Ενέργειας

Η ενσωμάτωση των ανανεώσιμων πηγών ενέργειας, και ιδίως της αιολικής ενέργειας, μπορεί να προκαλέσει διακυμάνσεις της ισχύος, σε ένα σύστημα ηλεκτρικής ενέργειας, λόγω της διαλείπουσας φύσης τους, όπως παρατηρείται στο σχήμα παρακάτω.



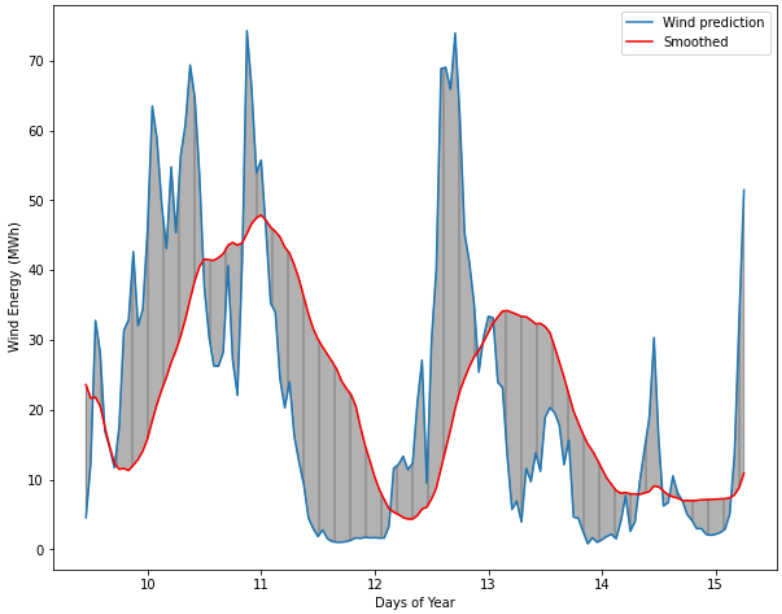
Σχήμα 6: Διακυμάνσεις της ωριαίας αιολικής ενέργειας στα πλαίσια πέντε ημερών

Προκειμένου να χρησιμοποιηθεί αιολική ενέργεια στο δίκτυο ηλεκτρικής ενέργειας παρά τις διακυμάνσεις της, απαιτείται η εξομάλυνση της παραγωγής αιολικής ενέργειας είναι μια από τις καλύτερες δυνατές λύσεις που μπορούν να εφαρμοστούν για την επίλυση αυτού του προβλήματος. Η μέθοδος που αναπτύχθηκε στην διπλωματική αυτή για την εξομάλυνση της αιολικής ενέργειας είναι η εξομάλυνση μέσω battery energy storage

system. Με τη χρήση ενός «έξυπνου» αλγορίθμου προκύπτει το εξής αποτέλεσμα για την παραγόμενη τελική ενέργεια:

$$P_{sm} = \begin{cases} P_{wind} + P_{bat}^{d} \\ P_{wind} - P_{bat}^{C} \end{cases}$$

,όπου P_{sm} αντιπροσωπεύει την ομαλοποιημένη ενέργεια, P_{wind} την παραγόμενη ενέργεια από τις αιολικές γεννήτριες και P_{bat} την ενέργεια που διαχειρίζονται οι μπαταρίες, είτε φόρτισης είτε εκφόρτισης. Συνεπώς εφαρμόζοντας τον αλγόριθμο που αναπτύχθηκε παρατηρείται η εξής διαφοροποίηση στην ενέργεια, συγκριτικά με το άνωθεν διάγραμμα:



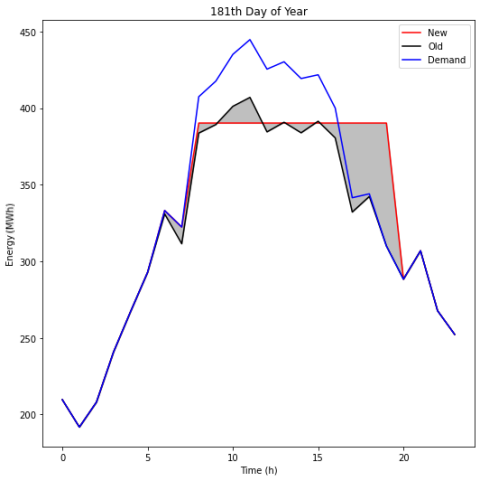
Σχήμα 7: Σύγκριση ομαλοποιημένης με παραγόμενη αιολική ενέργεια

Τα τμήματα που βρίσκονται πάνω από την κόκκινη γραμμή αφορούν την ενέργεια που «περισσεύει» και χρησιμοποιείται για φόρτιση της μπαταρίας και τα τμήματα που είναι κάτω από αυτή, την επιπλέον ενέργεια που απαιτείται για να παραχθεί αυτή η καμπύλη. Ως μέσω αποθήκευσης ενέργειας χρησιμοποιήθηκαν μπαταρίες Li-ion με συντελεστή of η_c και η_d ίσο με 85%. Συνεπώς η τελική χωρητικότητα που χρειάστηκε για τη λειτουργία του συστήματος είναι 780MWh. Με την ομαλοποίηση αυτή επιτεύχθηκε εξομάλυνση της καμπύλης παραγωγής της θερμικής ενέργειας. Επιπλέον, η εξομάλυνση της θερμικής καμπύλης θα βοηθήσει στην κατασκευή του αλγορίθμου για το συνολικό σύστημα αποθήκευσης ενέργειας. Τέλος, βοηθά σε άλλους τομείς. Μειώνει τη διαλείπουσα ισχύ της αιολικής ενέργειας, η οποία ονομάζεται «Ramp Rate» [36].

14.2 Αλγόριθμος Peak Shaving

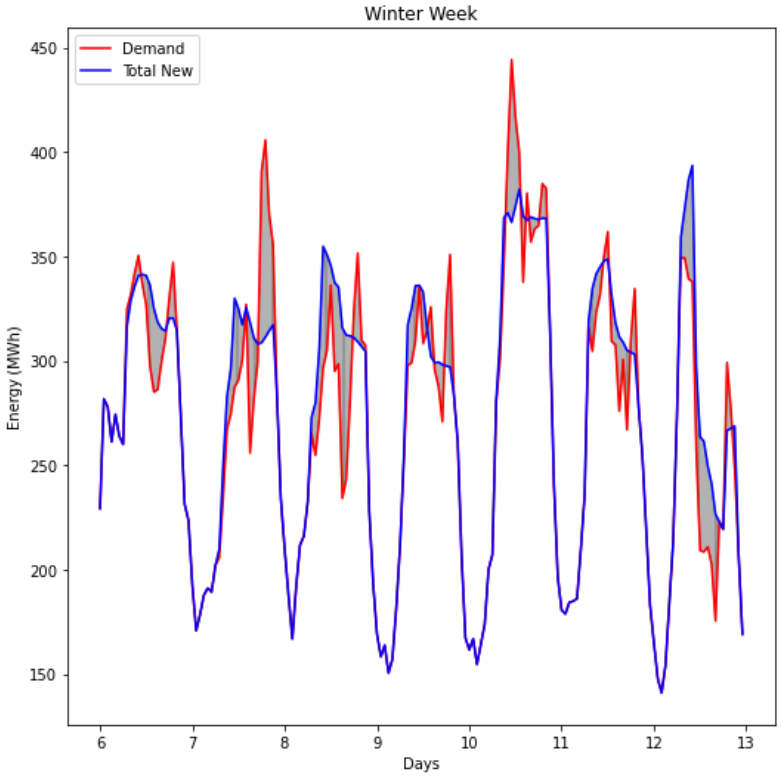
Η ιδέα πίσω από τον δεύτερο αλγόριθμο βελτιστοποίησης με τη χρήση ενός συστήματος αποθήκευσης μπαταρίας είναι πολύ διαφορετική από την προηγούμενη. Αυτή τη στιγμή, ο αλγόριθμος έχει την ευθύνη να εξοικονομεί ενέργεια το πρωί, να την αποθηκεύει και να τον χρησιμοποιεί τις νυχτερινές ώρες.

Αναπτύχθηκε αλγόριθμος ο οποίος έπαιρνε σαν δεδομένα εισόδου την πρόβλεψη του φορτίου, όπως αυτή προκύπτει από το νευρωνικό δίκτυο, την παραγωγή των PV της επόμενης ημέρας και την τιμή περικοπής της κορυφής. Με βάση τα προηγούμενα δεδομένα εισόδου και κατόπιν συγκεκριμένης διαδικασίας, έβγαζε σαν έξοδο την πορεία λειτουργίας των συμβατικών μονάδων παραγωγής καθώς και τις τιμές ισχύος που θα έπρεπε να αποθηκευτούν στο σύστημα μπαταριών (BESS) κάθε ώρα της ημέρας. Κατά την αρχικοποίησή του αλγορίθμου αυτού, ο οποίος εκτελείται για κάθε ημέρα του έτους, θεωρείται ένα επίπεδο ισχύος (που ονομάστηκε offset), πάνω στο οποίο προστίθεται η συνολική παραγωγή των PV, ενώ ταυτόχρονα περικόπτονται οι αιχμές λειτουργίας των μηχανών. Το επίπεδο offset, μεταβάλλεται (αυξάνεται σταδιακά) κατά την επαναληπτική διαδικασία του αλγορίθμου και ταυτόχρονα συμπαρασύρει μαζί του την καμπύλη παραγωγής των PV, δημιουργώντας έτσι μια νέα «συνθετική» καμπύλη συνολικής παραγωγής ενέργειας, η οποία κάποια στιγμή εμφανίζει σημεία τομής με την καμπύλη του φορτίου της ίδιας ημέρας. Η διαδικασία αυτή συνεχίζεται περεταίρω με τον ίδιο τρόπο, μέχρις ότου να δημιουργηθεί μια «τεχνητή» περίσσεια ενέργειας, η οποία αποθηκεύεται στο σύστημα μπαταριών και προσεγγίζει με την ελάχιστη θετική διαφορά, το ποσό ενέργειας που θα χρειαστεί για να καλυφθεί η αιχμή του φορτίου που θα ακολουθήσει αργότερα την ίδια ημέρα. Σε ορισμένες περιπτώσεις η καμπύλη παραγωγής ενέργειας παρέμενε σταθερή και περισσότερη από την απαιτούμενη ζήτηση ενέργειας, καθώς όταν ήταν αναγκαίο λόγω αυξημένου φορτίου των επόμενων ημερών λειτουργούσε και ως επιπλέον γεννήτρια της μπαταρίας. Αυτό μπορεί να γίνει εύκολα κατανοητό από το ακόλουθο σχήμα που αντιπροσωπεύει την λειτουργία μιας καλοκαιρινής ημέρας.



Σχήμα 8: Τελική καμπύλη παραγωγής ενέργειας σε σύγκριση με την ομαλοποιημένη και τη ζήτηση

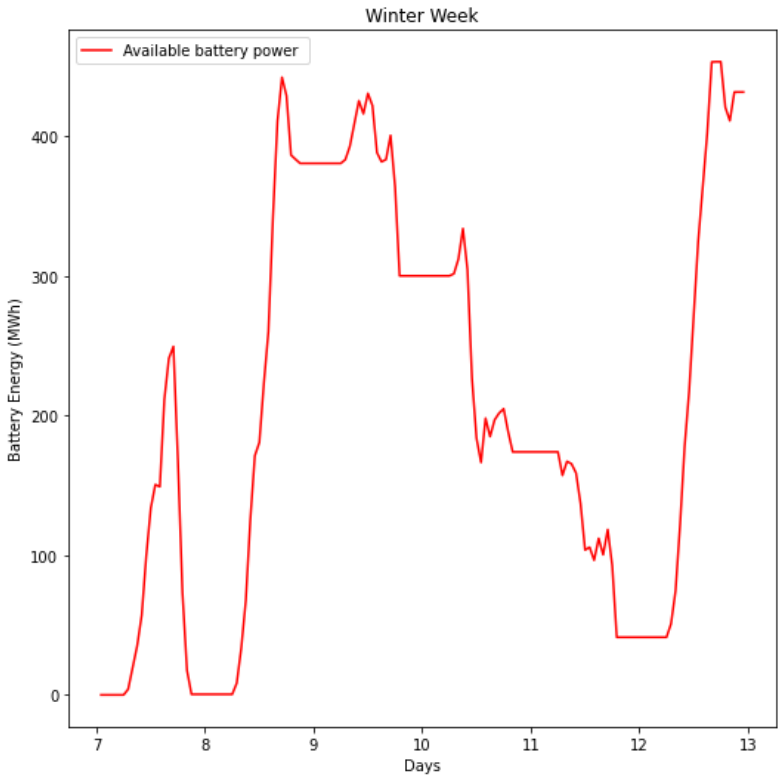
Από τις παραπάνω δράσεις προκύπτει ένα τελικό νέο μοντέλο παραγωγής ενέργειας για το νησί. Το αρχικό μοντέλο αποτελούνταν από τις τρεις μορφές παραγωγής ενέργειας και τις ενεργειακές ανάγκες του νησιού. Το νέο μοντέλο διαθέτει δύο επιπλέον συστήματα BESS. Ο τρόπος για να προσδιοριστεί η απαιτούμενη χωρητικότητα μπαταρίας για την BESS είναι να βρεθούν οι περιοχές φόρτισης και εκφόρτισης κατά το έτος που μελετάται. Μια εβδομάδα αναφοράς απεικονίζεται στο παρακάτω σχήμα.



Σχήμα 8: Παράδειγμα χειμερινής εβδομάδας όπου απεικονίζεται η νέα παραγόμενη ισχύς σε σχέση με την ζήτηση

Στη δεδομένη εικόνα, η μπλε και κόκκινη γραμμή αντιπροσωπεύουν τη νέα παραγόμενη ενέργεια και τη ζήτηση ενέργειας αντίστοιχα. Οι επισημασμένες περιοχές κάτω από την μπλε γραμμή και πάνω από την κόκκινη γραμμή είναι η ενέργεια φόρτισης και οι περιοχές κάτω από την κόκκινη και πάνω από την μπλε γραμμή είναι οι περιοχές εκφόρτωσης της μπαταρίας. Από τον υπολογισμό αυτών των περιοχών, για όλη τη διάρκεια του έτους προκύπτει το συμπέρασμα πως η αναγκαία χωρητικότητα μπαταρίας είναι 700MWh. Στο

ακόλουθο σχήμα, φαίνεται η διαδικασία φόρτισης και εκφόρτισης της μπαταρίας για τη δεδομένη εβδομάδα.



Σχήμα 9: Παράδειγμα χειμερινής εβδομάδας όπου απεικονίζεται η διαδικασία φόρτισης και εκφόρτισης της μπαταρίας

15. Σύνοψη – Συμπεράσματα

Καθώς η κοινωνία αυξάνεται συνεχώς και διευρύνει τη ζήτηση για ενέργεια και η κατανάλωση ενέργειας αυξάνεται επίσης. Με την τεχνολογική πρόοδο που έχει σημειωθεί, οι ανανεώσιμες πηγές ενέργειας αποτελούν σημαντικό μέρος της υποδομής του ηλεκτρικού δικτύου. Με οποιαδήποτε ευκαιρία ανάπτυξης μπορεί να υπάρξουν θετικές και δυσμενείς επιπτώσεις. Η διπλωματική αυτή έχει ως στόχο να παρουσιάσει αξιόπιστα και ακριβή μοντέλα για την πρόβλεψη των παραγόμενων ενεργειών από μορφές ανανεώσιμων πηγών ενέργειας και να δώσει μια λύση για τον μετριασμό ζητημάτων ηλεκτρικού δικτύου που προκύπτουν με υψηλά επίπεδα διείσδυσης PV και αιολικής ενέργειας σε τροφοδότη διανομής. Αναπτύχθηκαν νέες μορφές αλγορίθμων πρόβλεψης δεδομένων και τεκμηριώθηκαν με ακρίβεια οι λόγοι που επιλέχθηκε ο καθένας από αυτούς σε κάθε περίπτωση.

Όσον αφορά τη διαχείριση ενέργειας, εισήχθη και ενσωματώθηκε ένα προγνωστικό σύστημα διαχείρισης ενέργειας (EMS) με βάση την πρόβλεψη φορτίου και ενσωματώθηκε στη λειτουργία ενός συστήματος αποθήκευσης ενέργειας μπαταρίας (BESS). Σε αυτή τη διατριβή, παρουσιάστηκαν δύο διαφορετικά μοντέλα BESS. Ο πρώτος αλγόριθμος αναπτύχθηκε για τη σταθεροποίηση της παραγωγής αιολικής ενέργειας. Ο παράγοντας που επηρεάζει την παραγωγή αιολικής ενέργειας οφείλεται κυρίως στην ταχύτητα και την κατεύθυνση του ανέμου, δύο παράγοντες που είναι έντονα μεταβλητοί. Συνεπώς, είναι λογικό να απαιτηθεί η σταθεροποίηση της παραγόμενης ενέργειας. από την άλλη πλευρά, η ηλιακή ενέργεια είναι γνωστό ότι παράγεται μόνο κατά τη διάρκεια της ημέρας. Επιπλέον, έχει πολύ μικρότερες στιγμιαίες και οριακές διακυμάνσεις σε σχέση με την αιολική ενέργεια, όπως και παρόμοια παραγωγή κατά τις ίδιες περιόδους σε σύγκριση με τα έτη. Για το λόγο αυτό, επιλέχθηκε μια μορφή αποθήκευσης επιπλέον ηλιακής ενέργειας που δεν καταναλώνεται για τη μείωση του μέγιστου φορτίου ενέργειας που παράγεται από τον θερμικό κινητήρα.

Τα αποτελέσματα των προσομοιώσεων έδειξαν ότι με την εφαρμογή της προτεινόμενης μεθοδολογίας, είναι δυνατή η επίτευξη ομαλότερης λειτουργίας των θερμικών μηχανών καθώς ακόμη και η αντικατάστασή τους από μηχανές μικρότερης ισχύος, ενώ ταυτόχρονα βελτιώθηκε η αξιοποίηση της παραγόμενης ηλεκτρικής ενέργειας από το φωτοβολταϊκό πάρκο, επιτρέποντας μεγαλύτερη διείσδυση τις νυχτερινές ώρες αιχμής.