International Journal of Engineering, Management and Humanities (IJEMH)
Volume 3, Issue 6, pp: 185-194 www.ijemh.com

Wind Energy Analysis and Forecast using Machine Learning

Mudassir Hussian Magray[1], Sonal Sood[2]
[1]M. Tech Scholar, Department ofElectrical Engineering, Rayat Bahra University, Chandigarh, India
[2]Associate Professor, Department of Electrical Engineering, Rayat Bahra University, Chandigarh, India

Date of Submission: 12-12-2022 Date of Acceptance: 27-12-2022

Abstract: A key and expanding field in the renewable energy sector is wind energy prediction. Given that biofuels are absorbed into existing programs and integrated with conventional sources, determining the amount of energy that will be delivered is crucial to lowering the wind farm's production costs and assuring the safety of grid access. In this research, we'll apply several cutting-edge Deep Learning techniques to a variety of machine learning problems, including the development of a prediction model for wind energy output.Convolutional neural networks and deep, fully integrated layer perceptron networks that may profit from the spatial and hierarchical aspects of statistical current weather patterns will both be taken into account. We'll also explore the consequences of regularization techniques like weight decay and dropout, and consider which are the best predictive deep models after analyzing the development of their retraining.

Keywords: Forecasting, Wind Energy, Deep learning

I. INTRODUCTION

Multilayer Perceptrons (MLPs) started to undergo a little decline about 1990 after experiencing a big initial impact. Even while the deep residual computation of the gradient of the MLP error function could be carried out in a pretty straightforward manner, the challenge of designing successful MLPs with three or more layers was a particularly puzzling one. The vanishing gradient issue [8], which was partially brought on by insufficient weight initialization, was the root cause of this..

This radically changed after G. Hinton and R. Salakhutdinov published their seminal study [12], which showed how an unconstrained layering scheme based on Boltzmann machines could create a satisfactory indexing (or pretraining) of the weights of a many-layered MLP. The model was then effectively improved via classifier. Soon after, Y. Bengio and his associates [4] presented a comparable but marginally simpler pretraining method based on stacked machine learning algorithms. The interest in deep MLPs, or MLPs with three or more layers, and deep learning in general, has recently increased as a result ofthis.

As a result of this focus, the initial Hinton and Bengio schemes have been significantly simplified, and numerous new theories and techniques have been introduced to the MLP field. These include new classifiers and the repairs of some of the original MLP recipes, such as sigmoid installations or weight decay regularization, with new concepts like linear activation unit (ReLU) activations [9] or underachievement regularization [18]. Furthermore, because of the enormous datasets and deep MLP parameters, batch learning is generally not a possibility. As a result, many studies have focused on the selection of learning rates (or how to avoid them) or impetus techniques like Nesterov's acceleration. Online education frequently takes place over minibatches of randomly selected sequences.This raises the issue of when to stop training, which was fairly easy in the batch training of conventional MLPs if the right regularization and an efficient optimizer were used. A variety of initialization strategies have been proposed that enable the training of successful deep models once the preconditions are met, making the need for specialized (and expensive) pretraining less pressing. A good example of such a complete approach is [19]. A purely convolutional layer is combined with a convolutional layer. Another crucial component of effective deep learning systems is this layer, which examines inputs utilizing localized pane filters with a pooling sublayer that pools the outputs of the preceding sublayer..Thanks to processing developed by Y. LeCun in the late 1990s, convolutional convnets have achieved cutting-edge results in tasks like MNIST [7] or ImageNet [14]. This processing is especially natural when inputs possess a spatial structure, as it does with photographs.

It is now possible to train very large deep nodes with dozens of weights successfully because to all these advancements, hence it is crucial to use applications that can take advantage of high-performancehardware with parallel computing (i.e., multicore processors) and vectorization (i.e., GPU units).

Due to the field's quick rate of growth and the absence of a recognized multipurpose design, it is also advised to rely on publicly accessible libraries and environments like the Pylearn2-Theano libraries [5] [3] [10] that we use or the Caffe [13] deep learning framework.

Whatever the case, it seems that the primary areas of attention for research into deep learning include computer vision, speech recognition, and problems with natural language recognition. This is at least partially predictable given the striking parallels connecting deep learning architectures and the processor systems in the visual cortex [13]. More and more, deep learning algorithms are being seen as pattern matching processes that generate higher abstract concepts at each layer so that characteristics in the upper layers capture perhaps more powerful concepts..

However, a strategy like this could work well for smaller regression problems that still contain input patterns with a spatial structure. The goal of this research is to anticipate wind energy production. Spain is a world leader in wind energy, with a very high penetration rate, and can occasionally meet a sizeable amount of its electricity requirements. It follows that it is essential to provide accurate wind energy predictions, with the recommended models being Support Vector Regression (SVR) for large-scale forecasting and simple MLPs (typically at the farm level). These models receive their inputs from numerical weather prediction (NWP) systems like the ECMWF [1] and the GFS [2].A variety of weather variables are projected at certain locations using a rectangular grid that covers the target regions and reflects some foundation orographic model. Thus, one may interpret an area-wide NWP forecast as a set of feature maps (the various meteorological variables) with a spatial structure, much to how the RGB channels of an image correlate to feature maps with a two-dimensional structure (that of the underlying geography).

Convolutional networks are one of the models that will be considered in this study since, according to the previous plan, they make sense for generating wind energy prediction. Our main objective is to offer a method for developing models that can produce precise forecasts from the initial data with the least amount of pre-processing and specialist knowledge. As a result, we must explicitly decide in advance on network initialization, online training, activation function, and regularization approach given the enormously large array of alternatives in the literature. Naturally, a more or less universal network architecture must also be adopted in addition to this.

We will go into more detail about our choices in the next sections. In addition to the standard "small" MLPs and SVR models that we use as reference performance metrics, we will consider deep MLPs with a standard multilayer structure, general deep cnn networks (CNNs), as well as an ability to adapt of the well-known LeNet [6], one of the most effective hardware platforms for computer vision. We'll employ Glorot-Bengio weight configuration [8, ReLUs as ann [9], attrition denoising [8,] load rating rotting in the final thoroughly layers of all of those thick nets, conjugate gradient as the classifier, random mini-batch autoencoder over batches of moderate size, and random mini-batch back propagation over batches of smaller sizes.As a result, we may operate with a constant, generally applicable learning rate that is no longer an explorable parameter. We review some of the most current DNN concepts and provide general recommendations for using deep MLPs to regression problems.

We thoroughly investigate the potential applications of the two main DNN paradigms to the problem of local and worldwide wind energy projection.

For the purpose of predicting wind energy, we provide a modification of the well-known LeNet convolutional neural network and show that it is super competitive with other DNN designs or modern methods like Support Vector Regression.

We will use the Pylearn2 [10]-Theano [3] [5] platform, as mentioned earlier, because it provides a wide range of neural networks that have been thoroughly tested and allows us to study many of the most current and effective ideas for deep network training. Given the web sizes and input attributes we work with, we can benefit from Theano's skills for program execution on GPUs. One of the key advantages of adopting Theano as the base numerical library is this. We run our testing on a machine with an NVidia Tesla K40 GPU, enabling us to investigate a significant number of deep modeling setups with suitable execution rates.

International Journal of Engineering, Management and Humanities (IJEMH)
Volume 3, Issue 6, pp: 185-194 www.ijemh.com

II. LITERATURE REVIEW

A multiple neural network was constructed by Catalao et al. The organization found that Levenberg-cascaded Marquardt's arrangement beat ARIMA and the permanence model for short-term prediction models. A dnn array method was also created for projecting renewable energy. [7]

Focken et al. [5] examined how regional filtering factors decreased the model complexity of combined generating electricity in the case of wind turbine aggregation. Scientists Pöller and Achilles [6] who study electromagnetics are mentioned. investigated the possibility of combining many breezes into one power. A more in-depth description of the methodology we utilized for our study's based on wind power prediction was originally presented in [10], which was followed by an introduction to it.

III. METHODOLOGY

In its early phases, machine learning needed enough computing capacity to function successfully across a variety of applications [2]. Because primary memory, space, and contemporary computers have become more efficient, handling larger amounts of data is now simpler.

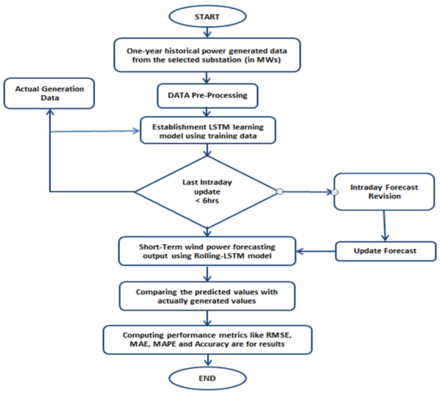

Figure 1 Forecast flow diagram

A. Wind Power Prediction

Numerous studies show that the utilization of wind energy has greatly expanded in recent years. The EU increased its renewable energy capacity by 12,800 MW in 2015. There are now 141 GW of megawatts available, of which 131 GW are located on land and 11 GW offshore [106]. With 45 GW, many European nations have the most installed capacity in the EU. As capacity grows, so does the need for projections that are more precise and reliable for a range of situations. This chapter provides an introduction to the spatiotemporal explanatory variables used in this research, covers the use of regulatory technology methodologies, and makes predictions about renewables..

B. Use Strategies for Predicting Electricity Generation

Precise duration predictions are necessary for many use scenarios. The distance that the power or wind strength value must be projected in front of the current time point t0 is specified by the projection horizon t.

The sale of wind and solar energy in the electricity markets, which can only be effective with correct estimates since the gas has already been distributed, makes substantial use of forecasts. The electromagnetics industry appears to have a trend toward shorter prediction horizons. For all day sectors of the European Voltage Source converter, the limit has been decreased to 30 minutes..

Table 1 for a characterization of distinct success

Horizon	Time Range	Applications
Very short-term	Few seconds – 30 minutes	Market clearing Trading Balancing Virtual Power Plants
Short-term	30 minutes – 6 hours	Load balancing Intraday trading Regulation
Medium-term	6 hours – 1 day	Day-Ahead trading Price optimization
Long-term	1 day – 1 week or more	Planning of reserve energy Scheduling of maintenance

Spot1. In the hydrocarbons trading industry, prediction periods ranging from minutes to days are typical for interday and overnight forecasts.

Grid realignment also needs precise estimations. Making early choices on reserve power and redistribution is necessary to fulfill the important aim of preserving grid voltage and frequency stability. Stakeholders include system managers and any energy source that injects electrons into the grid. Plans for electrical balancing also call for precise calculations. Control energy, also known as balances electricity, manages unforeseen grid events. In the European Union2, there is a cost for the public's access to regulations..

In the future, storage will become much more valuable. When should batteries or other energy storage devices be charged, and when should energy be used? The building of synthetic

International Journal of Engineering, Management and Humanities (IJEMH)
Volume 3, Issue 6, pp: 185-194 www.ijemh.com

power plants, which are composed of several types of power trees and bushes yet function as a single unit, also makes use of forecasting.
Table 3.1 categorizes the various forecast periods and includes usage examples. The methods that are currently practical for these projections are described in Section 3.2.

C. State of the Art

Numerical weather prediction (NWP) models and forecasts based on historical time series fall into two categories [22,29]. Both sectors employ a wide variety of methodologies and hybrid processes. An examination of wind power estimation techniques is provided by Foley et al. in [29].

A review on predicting produced power and wind direction was published by Lei et al. [3]. The detailed analysis of the methodologies by Soman et al.[12] focuses on their potential applications across a range of scopes. The numerous wind power forecasting techniques are characterized by Wang et al. [9].

1. Numerical Weather Prediction

Using physical estimates to characterize the climatic system, such as radiation, dispersion, and pressure values, NWP models are created. Navier-Stokes calculations are commonly used to explain the motion of liquid droplets because of regular principles [9]. Models with varying capacities, whether regional or worldwide, can be used to forecast the weather. The National Weather Prediction Program (NWP) is utilized in order to cope with fine spatial resolution and enhance atmospheric process models..

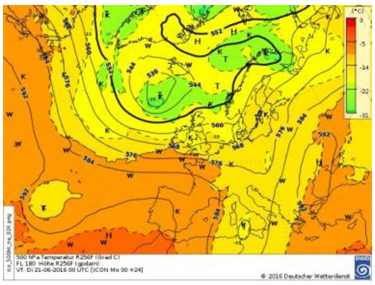

Figure 2 The German Accuweather gives examples of NWP.3

At research and weather service companies, models are computed on quantum computers. Forecasts are made for a range of purposes in study and application, as opposed to being created for a specific purpose [9]. The model generates a rough grid of predictions that account for both the wind speed and direction as well as the regional atmospheric conditions. The wind speed of the chosen turbine is calculated using a grid with grid points that span a few kilometers [9]. Using the power curve of the turbine, the speeds are translated into a power value (see Figure 3.2)..

These models have a variety of flaws while being frequently utilized. The weather systems that make them publicly available determine what may be expected. Forecasts are only available at specific times, and the time periods that can be used are always fixed. Physical models find it exceedingly difficult to accurately anticipate the climate because it is chaotic; as a result, other methods, such as chance and statistic, perform better over short forecast horizons. The response surface method (rsm) (CFD) has lately gained in favor alongside these strategies. Using the model provided by Marti and colleagues [10], four-hour forecasting is feasible.

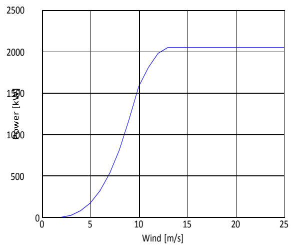

Figure 3. The power curve of an Enercon E-82 wind turbine5. For each wind speed, the expected power is given

In algorithmic meteorology, ensemble models are frequently employed for prediction [4]. On the one hand, anticipation may increase. Instead of using a predictable forecast, the Bayes theorem may be utilized to produce a prediction interval using uncertainty analysis. An ensemble forecast, which is a single prediction, is produced by combining many NWP models with varied starting points. Due of the dearth of trustworthy data on the status of the atmosphere in the realm of NWP models, this method is very useful. Even slight alterations to the original parameters have a major impact on the expected outcome.

The mathematical calculations of these cluster projections are quite labor-intensive. noteworthy development Gneiting et al. [6] developed communities using the ensemble model output statistics (EMOS) approach to account for different local weather variations and decrease prediction accuracy. Similar techniques are used by

Thorarinsdottir and Gneiting [10] to calculate the average wind speeds over the American Pacific Northwest..As Mahoney et al. [8] show, combining several NWP techniques into an ensemble can be quite beneficial. The emphasis on innovation placed by the NWP batch is one of the hypotheses employed. fitted using an analog batch Kalman filter and a regression. Junk et al. [12] present an analysis of a number of extension options using the example of an analog quintet. A system is used to generate both the forecast and the information regarding uncertainty.

2. Statistical Learning

It has now been established that data mining approaches work well for predicting wind speed and skill. Neural networks [2] and k-NN [12] are examples of effective systems. When given geographical data, machine learning techniques can offer reliable prediction results.

Support vector classification beats numerical climatologists for the shortest timeframes, according to Treiber et al. [11]. Based on projections from weather data in Germany, neural networks beat post-processed hydrological models by up to three hours (DWD). More than six hours should be spent using environmental measures. We want to enhance suggestions for this time period because there are numerous programs for short-term forecasting and the data-driven approach has several benefits. In the future, a variety of longer time horizons may benefit from hybridization with meteorological approaches. The sections that follow discuss the most important ml methods and their results.

3. Support Vector Techniques

The SVM approach is really useful. advantageous for modeling and classification applications. In instance, improved prediction skills can be achieved. The strategy serves as the foundation for our recommended solution and is covered in detail in Chapter 5. The SVR algorithm was used by Kramer and Gieseke [12] to predict short-term wind power with success. This study analyzes loss function variation, and evaluations of the projection at the grid point and park levels show that SVR appears to perform well when -loss6 is used..

Mohandes et al. [13] also determine wind speed using SVM. Performance of SVM prediction is comparable to MLP learning methods. Generally speaking, the SVM model is preferable to the learning approach. On the other hand, the findings are restricted to the Madina, Saudi Arabia, mean day dataset and do not offer data for short-term planning horizons..

IV. SYSTEM IMPLEMENTATION

In this part, we'll use machine learning to forecast wind energy output, first over the Spanish peninsula and subsequently on the Sotavento wind farm.

A. NWP and Production Data

The European Centre for Medium-Range Weather Forecasts (ECMWF) system's eight forecast for weather variables that we will use for weather forecasts (NWP) are as follows:

- P, the pressure at surface level.
- T, the temperature at 2m.
- V_x, the x wind component at surface level.
- V_y, the y wind component at surface level.
- V , the wind norm at surface level.
- V_x^{100}, the x wind component at 100m.
- V_y^{100}, the y wind component at 100m. – V^{100}, the wind norm at 100m.

The input dimension in the Sotavento instance is therefore 1598 = 1,080 since the measurements are made on a 15 x 9 rectangular grid that is centered on the Sotavento location (43.34N, 7.86W). With a very high input size of 57 35 8 = 15,960, we now explore a 57 x 35 rectangular grid that completely encloses the Iberian peninsula for peninsular Spain.

The general public can get information about Sotavento's wind energy; Peninsular Spain's information was kindly provided by Red Electrica de Espa?a (REE). In order to normalize them to the [0,1] range, we divide the actual wind energy output by the greatest attainable value in each situation. Our study will make use of data from the years 2011, 2012, and 2013, which will be utilized as the training, validation, and test subsets, respectively. Because NWP predictions are released every three hours, each subset will typically contain (24/3) 365 = 2,920 patterns.

B. Deep Models

We'll examine deep networks that have all of their layers completely linked, which we call deep MLPs, as well as deep convolutional neural networks, often known as deep CNNs, which feature a number of initial convolutional layers followed by completely integrated ones. We will employ Algorithm 1 Over search using Support Vector Regression (SVR) products and "standard" one hidden layer MLPs as past approaches..

International Journal of Engineering, Management and Humanities (IJEMH)
Volume 3, Issue 6, pp: 185-194 www.ijemh.com

procedure Hyper-parameter search(n,m) ▷ $n \times m$ iterations

$p^* = p$ ▷ p^*

randomly initialize an hyper–parameter vector p

: optimal hyper–parameter vector

for $i = 1,...,n$ **do**

for $j = 1,...,m$ **do**

$k \leftarrow$ random value in $\{1,...,m\}$

$p_k \leftarrow$ random value in $\{v_1^k,...,v_{N^k}^k\}$

evaluate the p–parameterized model and update p^* if needed

end for

end for

return p^* **end procedure**

. Because there are so many different alternative architectures and alternatives available to them, there would be an overwhelming number of model hyperparameters to research in order to find the best ones. In order to limit this as much as possible, we first set a couple of them to fair levels that resulted in respectable results in a first coarse model examination.

One such choice is the deep designs, which should be taken into account. For deep MLPs, we will consider two hidden layers with the same number of units. The standard deep CNN (sdCNN), our initial deep CNN choice, will begin with a convolutional layer and move on to two fully linked layers that each have the same number of units. For our second CNN alternative, which we call LeNet CNN or lnCNN, we will modify the well-known LeNet-5 architecture [16], which was developed specifically for the MNIST character recognition problem.

We'll employ non-symmetric ReLUs at the hidden layers and, as was previously discussed, we'll use the Glorot-Bengio heuristic from [8] to initialize the network by scaling up the initial weights by a factor of 1.5 and using a 0-symmetric uniform distribution with a width adjusted to the layers' fan-in.

Our training strategy will be conjugate gradient descent (CGD) across a set of random mini-batches. We used values of either 200 or 250, or about 6% and 9% of the learning sample size, because the size of the weights does affect the network's performance. In other words, we apply CGD over each new mini-batch starting with the loads established over the preceding mini-batch. Our error measure is the mean absolute error (MAE)

$$MAE = \frac{1}{N}\sum_{n=1}^{N}|D(x_n;P) - y_n|,$$

where D(x;P) denotes the depth of the current deep network D built using the pattern x's set of parameterized P. We employ the MAE rather than the more popular squared error because it represents energy deviation and, consequently, the energy to be lost or recovered from alternative generating sources to make up for errors in wind energy estimates...

As we'll see in the next section, using mini-batches during training results in sporadic spikes in the overall MAE evolution, which translates to the MAE values utilized for validation. In addition, validation MAE appears to be steady despite training MAE's ongoing fall. Since there is a minimum 1% reduction in MAE in the final 100 epochs, our svm classification method comprises training a deep NN for a maximum of 1,000 epochs (i.e., goes throught the entire training set). There are eight of these characteristics because we will create an input feature map for convolutional networks using each weather variable. Given the aforementioned possibilities, our sole alternatives are the following characteristics..:

- For deep MLPs, we must select the mini-batch size, weight decay and dumping indices, the number of hidden layers (one or two), the number of hidden units per plane, and the number of hidden layers overall (which we indicate by MLP2).

For the traditional deep CNN, we extend the deep MLP parameters by convolutional filter and pooling sizes, as well as their steps (which we refer to as CNN).

For the LeNet CNN (also referred to as LeNet), we must select the deep MLP parameters, but we may streamline the other choices by selecting the filter and sharing sizes and strides as properly scaled duplicates of the selections we made for LeNet-5.

In any case, it is clear that there are still too many hyperparameters for a full grid search, even with the earlier simplifications. We used a greedy approach to address this in which we first fixed the number of fully connected hidden layers at 2 and then performed Algorithm 1, which assesses models in terms of the MAE over the testing subset. Each of the algorithm's n = 50 external rounds examines a concrete hyper-parameter vector p.Hyper-parameters are taken into account for the fully connected layers' hidden unit count, weight decay multipliers used, dropout percentage, and given a dataset size. For each potential updating value pk from the list of values for the k-th the over to be investigated, m random hyper-parameter indexes are chosen on each external iteration. Both random choices are identical. The actual test results were

- Hidden unit numbers: 50, 100, 150, 200, 250, 300, 350, 400.

International Journal of Engineering, Management and Humanities (IJEMH)
Volume 3, Issue 6, pp: 185-194 www.ijemh.com

– Weight decay multipliers: 0.1, 0.2, 0.3, 0.4, 0.5.
– Dropout fractions: 0.3, 0.4, 0.5, 0.6, 0.7, 0.8.– Minibatch size: 50, 100, 150, 200, 250, 300.

For the deep CNNs, the stride was adjusted to 1, and the filter and pool sizes were modified using a constrained heuristic search. Be aware that these modifications entail at least four more parameters, making it almost impossible to do a complete random search of the subspace. Convolutional feature maps are available in an analogous number. The following are the provided deep NN values from the hyperparameter search.:

Table 2. Mean Absolute Errors for the Sotavento and REE problems

	MAE Sotavento			MAE REE		
	Test	Validation	Train	Test	Validation	Train
SVR	7.80	6.73	5.62	3.13	3.30	1.01
LeNet	7.63	6.25	5.82	3.13	3.01	2.48
CNN	7.76	6.26	5.39	3.31	3.05	1.96
MLP2	7.76	6.33	5.86	3.37	2.96	1.97
MLP1	8.25	6.41	5.51	3.70	3.10	1.81

The requirements for Deep MLPs (MLP2) for Sotavento are two hidden layers of 250 units each, a weight decay value of 0.3, a dropout coefficient of 0.7, and a mini-batch size of 200. The value decay and washout parameters, hidden layers layers, and 250-piece mini-batch for the REE variants will be the same.

For Sotavento, the typical deep CNNs (CNN) will contain a first convolutional layer with 2 6 layers and max pooling across 2 2 patches. Following this layer are two 200 unit fully linked layers with no weight decay, a 0.7 dropout factor, and a 250 unit mini-batch size. The topology of the REE CNN remains same as; the first layer now includes 3 3 filters, and maximum pooling is carried out over 3 5 patches. Following this are two 400 unit layers with 0.3 and 0.7 weight decay and washout coefficients and 200 unit mini-batch sizes. For Sotavento, we employed 16 convolutional maps, while for REE, we used 8.

A first convolutional layer with 2 2 filters and maximum pooling, as well as a second one with 4 2 filters and 2 2 max pooling, are both features of the LeNet-5 (LeNet) network that has been modified for Sotavento. Following them are two 200 unit layers with complete connections, no weight decay, a dropout coefficient of 0.7, and a 250 mini-batch size..Two convolutional layers make up the LeNet network for REE; the first layer has 68 filters and 22 maximum pooling, while the second layer has 66 filters and 22 maximum pooling. Two layers with 200 units each with dropout and weight decay values of 0.3 and 0.7 are placed after them. These layers are totally coupled. For both tasks, we used 32 and 16 convolutional feature maps in the first layer, respectively...

V. SIMULATION AND RESULTS

We also take into account a Gaussian SVR model and a "typical" 10-unit MLP with one hidden layer for comparison's sake. The SVR hyperparameters were determined via a grid search using the widely used LIBSVM software [6], and their best values were C=128.0, = 3.0518 105, 01 for REE, 0625 for Sotavento, C=128.0, and = 12.2078 105. For the traditional MLPs, we once more used Pylearn2-Theano, and the best Sotavento and REE parameters were 200 mini-batch sizes and 0.001 weight decay coefficient, respectively..

Table 2. The Sotavento (top) and REE (bottom) deep models' training complexity parameters and times in seconds

Model	#Params.	#Iters.	Time	Time/Iter.
LeNet	140808	426	1175	2.76
CNN	105736	500	705	1.41
MLP2	332750	259	276	1.07
LeNet	224776	949	19494	20.54
CNN	548176	717	6880	9.60
MLP2	4878300	258	1208	4.68

Table 1 displays the training, validation, and test errors for the two tasks and the top models. As can be seen, the SVR and LeNet-5 models outperform one another in the REE problem, with the second and third-place finishes going to the other two deep models. The standard MLP comes in last place. The LeNet-5 model, therefore, is by far the most effective model in Sotavento, followed by the SVR, the two deep models, and the ordinary MLP. We stress that although we assess models with a straightforward train-validation-test technique, a more accurate comparison should be made using an appropriate statistical test, such as the well-known Wilcoxon Rank Sum test, which takes into account more than simply MAE..

Figure 1 depicts the development of the train, test, and validation errors for the top CNN and LeNet-5 networks for Sotavento (top) and REE. (bottom). Although Sotavento's validating and test error history initially appears to be growing more smoothly, this is primarily due to a scale effect. Mini-batch training causes the enormous error oscillations (about twice as large for Sotavento than for REE). Although training error would continue to decrease because to the multiple classes of this problem, the lowest errors for Sotavento appear to

International Journal of Engineering, Management and Humanities (IJEMH)
Volume 3, Issue 6, pp: 185-194 www.ijemh.com

have essentially stayed at their current values. This is likewise true for the validity and test errors in REE..The vertical dotted line in both cases indicates the epoch with the lowest validation error, whereas the horizontal dotted lines suggest.

The deep models used to resolve the Sotavento (top) and REE (bottom) problems' complexity parameters and training times in seconds are listed in Table 2. All models are rather large, especially those used for REE, as can be shown (remember that input dimensions are respectively 1,080 and 15,960). Convolutional processes make feedforward passes and gradient backpropagation far more expensive for convolutional networks, even though they have fewer parameters than MLP2. We see that the smaller number of weights in LeNet for REE is caused by the larger filters used.

Deep training must therefore make use of all available hardware-based advancements, which is fairly expensive.

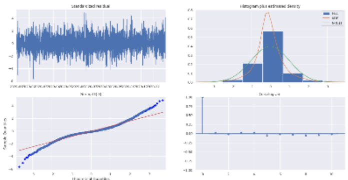

Figure 4 MAE and REE Matrixpredicting the next 24 steps

A. Plotting the predicted values

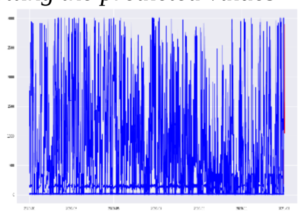

Figure 5 Predicted Values Plot

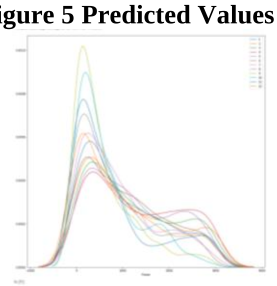

Figure 6. Power generation prediction per month

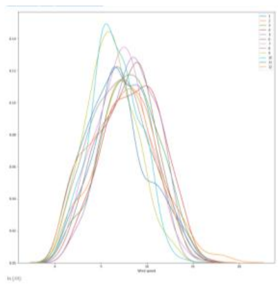

Figure 7 Wind speed prediction per month

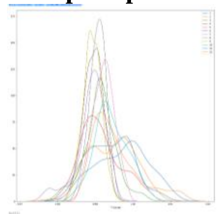

Figure 8 Wind pressure prediction per month

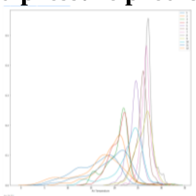

Figure 9 Air temperature prediction per month

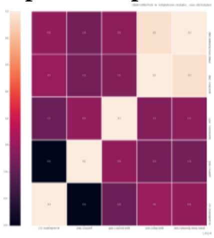

Figure 10 Confusion matrix

```
df=pd.read_csv('/Users/tahirshowkatbazaz/Desktop/Electrical datasets/wind.csv')
df.head()
```

Out[22]:

	Time stamp	System power generated | (kW)	Wind speed | (m/s)	Wind direction | (deg)	Pressure | (atm)	Air temperature | ('C)
0	Jan 1, 12:00 am	1766.64	9.926	128	1.000480	18.263
1	Jan 1, 01:00 am	1433.83	9.273	135	0.999790	18.363
2	Jan 1, 02:00 am	1167.23	8.660	142	0.999592	18.663
3	Jan 1, 03:00 am	1524.59	9.461	148	0.998309	18.763
4	Jan 1, 04:00 am	1384.28	9.184	150	0.998507	18.963

In [24]:

Figure 11 power generated through the year prediction

```
df.head()
```

Out[24]:

	Time stamp	System power generated | (kW)	Wind speed | (m/s)	Wind direction | (deg)	Pressure | (atm)	Air temperature | ('C)
0	Jan 1, 12:00 am	0.913107	0.812552	-0.219858	1.000480	-0.808069
1	Jan 1, 01:00 am	0.534272	0.604465	-0.137188	0.999790	-0.779471
2	Jan 1, 02:00 am	0.230803	0.409125	-0.054517	0.999592	-0.717679
3	Jan 1, 03:00 am	0.637583	0.664374	0.016343	0.998309	-0.697081
4	Jan 1, 04:00 am	0.477869	0.576104	0.039963	0.998507	-0.655086

In [25]:

Figure 12 Air pressure and temperature prediction for the whole year

VI. CONCLUSION

Deep neural networks are without a doubt quite powerful, but configuring and selecting the ideal hyper-parameters may be difficult. When calibrated properly, they may frequently produce outcomes that are superior than those of other classical models, as we have demonstrated here on two wind energy scenarios. Because meteorological variables are included, both problems have a bi-dimensional input structure and may thus be thought of as input channels. This might provide an explanation for why convolutional layers gave the best deep findings. While deep learning training demands a lot of processing, it also lends itself nicely to GPU usage and the huge speedups that GPUs offer..

In any circumstance, the work outlined here must be considered a first step. One major area of future study is to take into account various convolutional architectures, especially those of the AlexNet type ([14]). An easy solution is to attempt to reduce variance by combining multiple deep models (note that they naturally have a low bias). Given the high variability of validation during instruction, standard MLPs typically repeat training using different random phase transformations. However, a simpler, less expensive alternative is to select a set number M of the models with the smallest validation that were gained in a single training run as the ones used here..

A number of suggestions for architectures, model training, regularization, and stimulation have also been created as a result of the tremendous activity in deep learning. We are also pursuing some of these options.

REFERENCES

[1]. Abdelaziz A, Rahman MA, El-Khayat M, Hakim MA. Short term wind power forecasting using autoregressive integrated moving average modeling. Proceedings of the 15th International Middle East Power Systems Conference (MEPCON’12). 2012. pp. 208:1–208: 6.

[2]. Alpaydin E. Introduction to Machine Learning. MIT Press; 2009.

[3]. Barbounis TG, Theocharis JB, Alexiadis MC, Dokopoulos PS. Long-term wind speed and power forecasting using local recurrent neural network models. IEEE Trans. Energy Convers. 2006;21(1):273–84. https://doi.org/10.1109/TEC.2005.847954.

[4]. Burton T, Jenkins N, Sharpe D, Bossanyi E. Introduction to Machine Learning. John Wiley & Sons Ltd; 2011.

[5]. Cadenas E, Rivera W, Campos-Amezcua R, Heard C. Wind speed prediction using a univariate ARIMA model and a multivariate NARX model. Energies 2016;9(2). https://doi.org/10.3390/en9020109. URL: http://www.mdpi.com/1996-1073/9/ 2/109.

[6]. Chang W-Y. A literature review of wind forecasting methods. J. Power Energy Eng. 2014;2:161–8. https://doi.org/10.4236/jpee.2014.24023. URL: https://www. scirp.org/journal/PaperInformation.aspx?PaperID=44881.

[7]. Chen T, Guestrin C. Xgboost: a scalable tree boosting system. Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’16 New York, NY, USA: ACM; 2016. p. 785–94. https://doi.org/10. 1145/2939672.2939785. [44] Breiman L. Random forests. Mach. Learn. 2001;45(1):5–32. https://doi.org/10. 1023/A:1010933404324.

[8]. Chi Z, Haikun W, Tingting Z, Kanjian Z, Tianhong L. Comparison of two multi-step ahead forecasting mechanisms for wind speed based on machine learning models. 2015 34th Chinese Control Conference (CCC) 2015. p. 8183–7. https://doi.org/10. 1109/ChiCC.2015.7260941.

[9]. de Alencar DB, deMattos Affonso C, de Oliveira RCL, Rodriguez JLM, Leite JC, Filho JCR. Different models for forecasting wind power generation: case study. Energies 1976;10(2017). https://doi.org/10.3390/en10121976. pp. 1–1976: 27.

[10]. Dokuz AS, Demolli H, Gokcek M, Ecemis A. Year-ahead wind speed forecasting using a clustering-statistical hybrid method. International Conference on Innovative Engineering Applications. 2018. p. 971–5. CIEA'2018.

[11]. Dowell J, Pinson P. Very-short-term probabilistic wind power forecasts by sparse vector autoregression. IEEE Trans. Smart Grid 2016;7(2):763–70. https://doi.org/ 10.1109/TSG.2015.2424078. [10] Lima JM, Guetter AK, Freitas SR, Panetta J, de Mattos JGZ. A meteorological-statistic model for short-term wind power forecasting. J. Control Autom. Electr. Syst. 2017;28(5):679–91. doi: 10.1007/s40313-017-0329-8.

[12]. Drucker H, Burges CJ, Kaufman L, Smola AJ, Vapnik V. Support vector regression machines. Advances in neural information processing systems. 1997. p. 155–61

[13]. Dumitru C-D, Gligor A. Daily average wind energy forecasting using artificial neural networks. Proc. Eng. 2017;181:829–36. https://doi.org/10.1016/j.proeng.2017.02. 474. 10th International Conference Interdisciplinarity in Engineering, INTER-ENG 2016, 6–7 October 2016, Tirgu Mures, Romania.http://www.sciencedirect.com/ science/article/pii/S1877705817310597.

[14]. Ekstrom J, Koivisto M, Mellin I, Millar RJ, Lehtonen M. A statistical modeling methodology for long-term wind generation and power ramp simulations in new generation locations. Energies 2018;11:2442:1–2442:27. https://doi.org/10.3390/ en11092442.