International Journal on
"Technical and Physical Problems of Engineering"
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

| December 2023 | Issue 57 | Volume 15 | Number 4 | Pages 234-244 |
|---|---|---|---|---|

# WIND SPEED FORECASTING BASED ON SIX MACHINE LEARNING APPROACHES: DEVELOPMENT AND BENCHMARKING STUDY

**I. Tyass A. Raihani K. Mansouri**

*EEIS Laboratory, ENSET Mohammedia, Hassan II University, Casablanca, Morocco*
*tyass1@yahoo.fr, abraihani@yahoo.fr, khalifa.mansouri@enset-media.ac.ma*

**Abstract-** The current paper presents an extensive study on the development and comparison of different machine-learning approaches to achieve accurate wind speed prediction, which is crucial for renewable energy generation. Machine-learning methods are more effective than traditional statistical and physical methods when working with large datasets. Six different machine learning algorithms, including Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), Support Vector Machine (SVM), and the Extreme Learning Machine (ELM), were evaluated in this study using wind speed data from each quarter of the year and the entire year. The algorithms' performance is assessed using different error metrics and computational time. The study found that all the algorithms performed well with both big and small datasets, with the LSTM model showing the best performance in terms of evaluation metrics (*MAPE* = 0.068). The study also highlights that the ELM algorithm is more efficient in terms of computational time (2.77 s), and had the ability to learn quickly from a training dataset. Additionally, the article explores the impact of the ratio of data splitting on prediction performance, and emphasizes the importance of selecting an appropriate split percentage that balances project objectives, computational costs, and representativeness of the training and test datasets.

**Keywords:** Wind Speed Prediction, Time Series, Machine Learning, Forecasting Models, Errors Metrics, Computational Time, Data Splitting.

## 1. INTRODUCTION

As reported in the British Petroleum (BP) Statistical Review of World Energy 2021, renewable energy sources contribute to 11.7% of the global power generation, suggesting that the remaining 88% is generated from non-renewable sources [1]. This implies that about 88% of the world's power generation is provided by non-renewable energies such as oil, coal, gas or nuclear. In other words, the overwhelming majority of the world's energy consumption is supplied by unsustainable, polluting and exhaustible energies sources. However, the advantages of renewable energies are numerous; they are inexhaustible on the human time scale, environmentally friendly and safe.

Renewable energies sources emit very little greenhouse gases and unlike fossil fuels or nuclear energy, they do not use exhaustible materials like oil or uranium. Wind energy is widely acknowledged as a well-established and cost-effective renewable technology, second only to hydroelectricity. Due to its scalability, the utilization of wind energy is essential for the advancement of modern electricity generation strategies, capable of meeting both large-scale industrial and small-scale domestic needs. According to the BP Statistical Review of World Energy 2021, wind power is the largest contributor to renewable electricity generation, accounting for 173 TWh, followed by solar power at 148 TWh. However, wind energy faces challenges such as intermittency and susceptibility to variations in wind speed and direction, which limit its potential.

Indeed, the efficiency of a wind turbine installation depends strongly on the available wind potential at a given site, which depends on the cube of wind speed. Therefore, forecasting and assessing the wind potential is crucial for grid management and control, including determining the amount of energy produced, protecting the system from high speeds, and assessing the feasibility of building wind power plants at specific sites based on their potential.

In this sense, this paper attempts to develop and contrast various machine-learning approaches for predicting time series of wind speed data. To achieve this purpose, we implemented six ML prediction algorithms, covering the four main areas of ML, including Feed Forward Network (FFN), Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and Support Vector Machine (SVM). The impact of seasons and the database size on the relevance of these models was evaluated by testing them on each quarter of the year and the entire year's data. The main sections of this paper are organized as follows: The first section provides a literature review of ML approaches used for wind speed forecasting.

The second section outlines the methodology and describes the six machine learning algorithms used in this research. Dataset exploration and processing are discussed in the third section. The fourth section describes an hourly one-step-ahead wind speed forecasting based on six ML methods: Extreme learning machines (ELM), Convolutional Neural Network (CNN), Long Short-Term

Memory (LSTM), Bidirectional LSTM (BiLSTM), Gated recurrent units (GRU), and Support Vector Regression (SVR). Finally, in the last section, results are thoroughly analyzed and discussed. The performances of six algorithms were compared based on different errors metrics and on computational time required for each model.

## 2. RELATED WORK

Accurate wind speed prediction is a critical aspect of optimizing the design and operation of wind energy systems, which plays a crucial role in maintaining the stability and reliability of the grid, as well as efficient power distribution. This can enable grid operators to efficiently manage electricity supply and provide a consistent and reliable flow of electricity to consumers, ultimately leading to increased adoption of renewable energy and reduced reliance on fossil fuels.

According to the literature, three principal methodologies for predicting wind speed have been recognized, which comprise physical, statistical, and machine learning (ML) techniques. Physical models explore the factors that influence wind speed, such as temperature, pressure, and altitude, using physical concepts to model the variation in wind speed. Nonetheless, physical models exhibit lower precision in the presence of highly irregular wind speeds, rendering them more appropriate for long-range predictions or as input for statistical models. Statistical and AI-based approaches, on the other hand, predict wind energy as a stochastic process. Statistical models are based on probability theory and mathematical statistics, employing historical data sets to determine the correlation between the input parameters and the wind variable outcome. Models such as Autoregressive Integrated Moving Average (ARIMA), Bayesian models, and Kalman filter-based models are frequently used. In comparison, machine learning algorithms have been shown to perform better when dealing with large amounts of data [2]. These approaches train the model and forecast using time-series wind speed data. The Artificial Neural Network (ANN) is the most extensively used branch of machine learning, with the ANN model having self-adaptation and learning capabilities.

Thus, the prediction of wind speed through the utilization of machine-learning models has become a prevalent subject of research. Artificial intelligence techniques generally have the ability to handle nonlinear functions and exhibit strong self-learning capabilities [3]. Xiao and colleagues [4] studied the use of a self-adaptive kernel extreme learning machine (KELM) to improve the accuracy of wind speed predictions while reducing training costs. They also considered a selection of input parameters to improve the efficiency of the KELM model. Navas, et al. [5] developed a multi-layer perception neural network (MLPNN) to forecast wind speed at an altitude of 65 meters using wind speed data as input. According to [6] support vector machines (SVM) for regression outperformed multilayer perceptron neural networks (MLPNN) in predicting wind speed. Santamaria-Bonfil G, et al. [7] conducted an empirical test to forecast wind speed and power using support vector regression (SVR), and compared its performance to that of the persistence model and time series models such as AR, ARMA, and ARIMA, they found that SVR demonstrated improved performance compared to these models.

Recurrent neural networks (RNNs) are one of the most effective models for handling sequential data, specifically time series data. Nevertheless, traditional (RNNs) models have their own limitations. They are unable to adequately capture long-term dependencies in the sequence of input data and cannot address the issue of long-term dependencies. However, its variant deep learning model, Long Short-Term Memory (LSTM) is very skilled at handling the time series problem. This model was widely employed to predict wind speed; the outcomes were compared to autoregressive integrated moving average and traditional artificial neural network models, according to the comparison, the LSTM approach is more accurate [8]. Convolutional neural network (CNN), convolutional long short-term memory network (CLSTMN) and Wavelet decomposition (WD) technique were used in [9] to remove the effects of noise from the original data and increase the accuracy of wind-speed forecasts. Gated Recurrent Unit (GRU) is a different application of RNN, it is simpler to compute and implement that the LSTM. According to the results obtained in [10], the proposed approach using the GRU algorithm demonstrated high skill and narrower prediction interval. Recently, several studies have been conducted using hybrid AI models to predict wind speed.

In [11] SVM combined the predictions of multiple LSTMs to generate Wind Speed forecasts. In [12], wind speed predictions were made using a combination of specific weights with four distinct types of ANNs: Back propagation neural network (BPNN), Elman network, Wavelet neural network (WNN), and Generalized regression neural network (GRN). Previous studies have shown that AI-based models outperformed traditional statistical models in wind speed and power forecasting.

## 3. DATASET EXPLORATION AND PROCESSING

### 3.1. Data Collection and Work Process

The machine-learning algorithms were subjected to training, testing, and validation processes using the wind data profile collected from the Prediction of Worldwide Energy Resources (POWER) Data Access Viewer v2.0.0, in the period from 1 January 2020 to 31 December 2020, at "Abdel Halak Torres" in "Al Koudia Al Baida – Tetouen" (Latitude: 35° 45' 35.1, Longitude: -5° 41' 19.9'). The collected wind speed measurements were recorded continuously at a height of 50 m. There are 8784 observations of average wind speed, each taken over a 60-minute period. The data visualization is illustrated in figure 1, and Table 1 shows the basic information of wind speed time series for each quarter in 2020. Based on the obtained values, it can be observed that the second and last quarters exhibit stronger wind and a higher degree of variability in wind speed around the average. To start with, the database is divided into four time periods corresponding to the seasons of the year. Each season dataset is then farther split into two parts: the training part (70%) and testing part (30%).

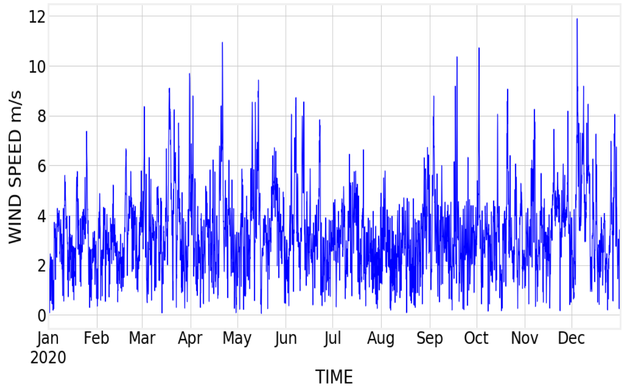



Figure 1. Wind data visualization

Table 1. Basic information of wind speed data (m/s)

| | Max | Min | Average | Variance |
|---|---|---|---|---|
| Quarter 1 - Winter | 9.690 | 0.080 | 3.185 | 2.310 |
| Quarter 2- Spring | 10.940 | 0.060 | 3.444 | 2.956 |
| Quarter 3- Summer | 10.360 | 0.090 | 3.013 | 1.803 |
| Quarter 4- Fall | 11.890 | 0.150 | 3.550 | 2.975 |

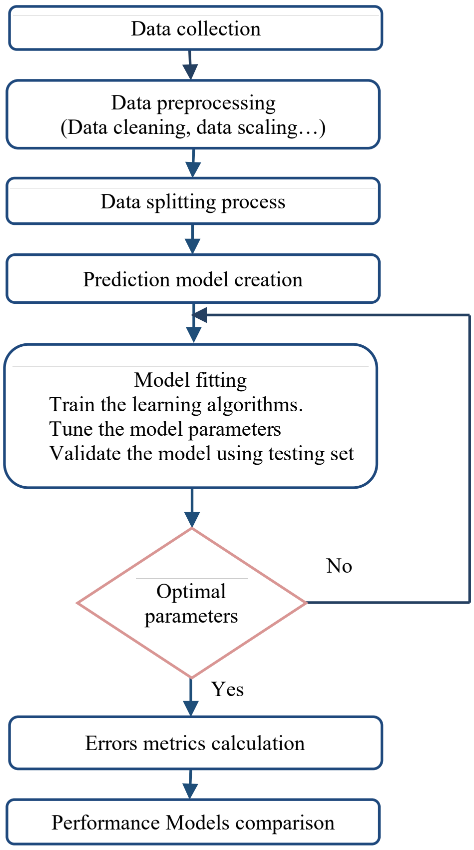



Figure 2. Overall Diagram of the proposed machine learning models

It is important to note that both the testing set and the training set use the same rolling prediction approach. Next, the machine learning models are trained to build the appropriate model. The obtained model was validated using the testing set, and was retrained until achieving a suitable accuracy, while also recording the final computational time. Multiple statistical error metrics, including mean square error (*MSE*), root mean square error (*RMSE*), mean absolute error (*MAE*), and mean absolute percentage error (*MAPE*), were employed to evaluate the prediction models. The same algorithms were applied to the entire database, i.e. the whole year, to assess the impact of the increased database size on the performance of the proposed prediction models. The main steps of the process used to perform predictive machine learning algorithms are depicted in Figure 2. Furthermore, several combinations of training and test data were applied to assess the effect of data portioning on prediction performance.

### 3.2. The Machine-Learning Algorithms

Arthur Samuel introduced the term ML in 1959, defining it as the research field that allows computers to learn without explicit programming. ML is a branch of AI that uses statistical probabilities methods that enable machines to learn or to accomplish tasks more efficiently based on previous information or data. The major objective of machine learning (ML) is to predict the value of an expected output given an input, using only characteristics supplied by the model programmer or learned from training data. Machine learning can be categorized into three broad categories: (i) supervised learning: The goal is to train the machine to identify the fundamental relationship between inputs and outputs, thus which supervised machine learning can be applied to forecast future data. The algorithm aims to generate a function that can effectively anticipate the output from the input variables with precision. There are two types of supervised machine learning, classification and regression. (ii) Unsupervised learning: The algorithm itself determines the structure of the input. Two categories of supervised machine learning exist, clustering and association. (iii) Reinforcement learning: In reinforcement-based machine learning, a computer program engages with an active environment to achieve a certain goal.

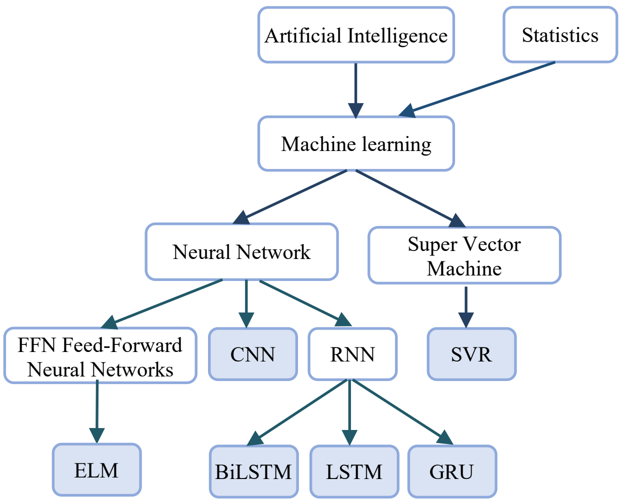



Figure 3. A schematic overview of the main branches of machine learning (ML)

As the learner-program moves through the problem space and adapts to the given context, it receives feedback in the form of "rewards" and "punishments" that help it identify the most effective behavior. Reinforcement learning is frequently applied in interactive or dynamic environments, such as gaming. Figure 3 shows a schematic overview of the main branches of machine learning (ML) methods including machine-learning model developed in this research, namely; the ELM, CNN, LSTM, BiLSTM, GRU, and SVR.

### 3.3. Evaluation Indices for Forecasting Performance

We used three statistical errors metrics to validate the developed algorithms and compare their forecasting accuracy, namely: *RMSE* (root mean square error) *MAE* (mean absolute error), and *MAPE* (mean absolute percent error), the calculation of these metrics is performed using the following formulae:

$$RMSE_{(y,\hat{y})} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2} \quad (1)$$

$$MAE_{(y,\hat{y})} = \frac{1}{n}\sum_{j=1}^{n}| y_j - \hat{y}_j | \quad (2)$$

$$MAPE_{(y,\hat{y})} = \frac{1}{n}\sum_{j=1}^{n}| \frac{y_j - \hat{y}_j}{y_j} | \quad (3)$$

where, $y$ and $\hat{y}$ are respectively the real and the forecast values and $n$ is the samples number of the $y(t)$ series.

## 4. FORECASTING ALGORITHMS DEVELOPMENT

### 4.1. Extreme Learning Machine

Extreme learning machine (ELM) is a feedforward neural network (FNN) using a single hidden layer. The nodes can be never renewed; they may be inherited from their ancestors without being altered, so it is not required to adjust the parameters of hidden nodes. The (ELM) algorithm exhibits high generalization performance with an extremely fast learning speed, faster than networks trained using backpropagation. Figure 4 illustrates the diagram representation of ELM model [13].

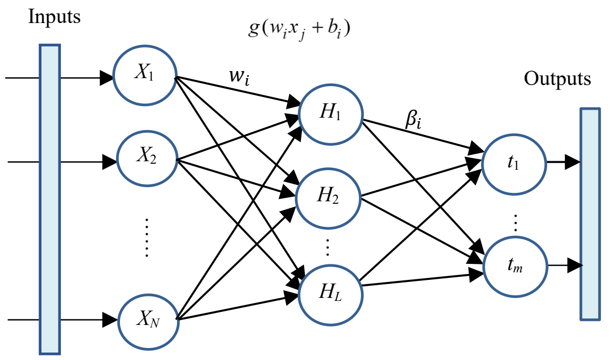



Figure 4. The diagram representation of ELM model [13]

The indices $N$, $L$ and m correspond respectively to the number of nodes in the input, hidden, and output layer, the activation function's output is represented as follows:

$$\sum_{i=1}^{L}\beta_i . g(w_i . x_j + b_i) = o_j \quad , \quad j = 1, \dots, N \quad (4)$$

where, $w_i$ and $\beta_i$ denotes the weight vectors, and $b_i$ represents the bias function. The goal is to minimize the error:

$$\sum_{j=1}^{L}\left\| o_j - t_j \right\| = 0 \quad (5)$$

To build the ELM model, we develop the following algorithm:

- Assign randomly the parameters ($w$, $b$) of the hidden layer from a Gaussian distribution with a range of -1 to 1.
- Compute the output matrix of the hidden layer by applying the Rectified Linear Unit (ReLU) as the activation function.
- Calculate the output weights $\beta$, following the equation

$$\beta = H^{+}T \quad (6)$$

The matrix $H^{+}$ refers to the Moore-Penrose inverse of the hidden layer matrix $H$. The purpose is to reduce the least square error between training and predicted variables

- The model is ready for forecasting, we use the test data to make the predictions.
- Using the performance metrics to assess the accuracy of predictions.

### 4.2. Convolutional Neural Network

In the literature, CNNs have primarily been used for image classification [14]. Using CNNs to predict the following value in a sequence issues, such as time-series forecasting problems, has recently attracted increasing interest from the research community [15]. CNNs employ a convolution, a specific linear operation. CNNs aim to learn the relation between inputs data and the output, and then save that learning in the weights of filters. Our wind speed dataset is a univariate time series, so we use a one-dimensional convolutional layer. The figure 5 shows a simplified diagram of a 1D CNN [15].

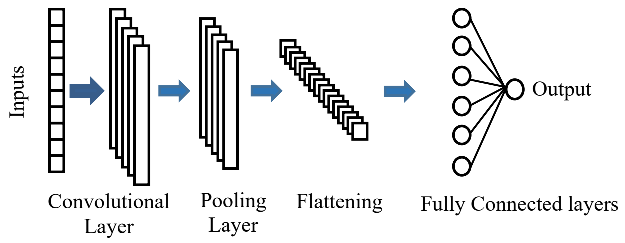



Figure 5. The diagram of 1D CNN [15]

The following is a description of our process for creating the 1D CNN model for one-step wind speed forecasting:

➢ Convert the one-dimensional time series into sub sequences by extracting features and outputs.
➢ Transform the inputs data in a 3D matrix format to be adapted to CNN models.
➢ Build the CNN model architecture: Firstly, we implement a 1D convolutional layer using a filter, the input data is convolved. Variables in the filter are analogous to neural network weights. The layer calculates the total of n weighted inputs time steps, app ends a bias, and after

applies the ReLU activation function, the corresponding equation is expressed as follows:

$$y_0 = \mathrm{ReLU}(w_0 x_0 + w_1 x_1 + \ldots + w_i x_i + b) \quad (7)$$

where, $y_0$ is the first output of the $1D$ convolutional layer, $x_i$ are the inputs, $b$ is the bias, and $w_i$ are weights. The identical process is applied for the subsequent time step:

$$y_1 = \mathrm{ReLU}(w_0 x_1 + w_1 x_2 + \ldots + w_i x_{i+1} + b) \quad (8)$$

➢ While maintaining the significant properties that the convolutional layer determined, the pooling layer control the overfitting of the parameters. Finally, fully connected layers forward the features obtained to output layer using one neuron.

➢ To tune the model, we use the (Adaptive Moment Estimator) algorithm to optimize the gradient descent problem and the mean squared error as loss function.

➢ In the last step, we train the model and test it on the test data.

### 4.3. Long Short-Term Memory

The LSTM network is a type of recurrent neural network (RNN). The main objective of LSTM is to resolve the gradient vanishing problem. Therefore, in LSTM, self-connected gates are incorporated in the hidden unit. The LSTM cell comprises three gates, namely the input gate $x_t$, forget gate $f_t$, and output gate $o_t$, which control the flow of information. There are also two types of outputs: Cell state $c_t$ and hidden state $h_t$. Figure 6 shows the LSTM cell structure. The inputs of each gate have different weights $w_i$ and biases $b_i$. The Forget Gate is capable of discarding information or reducing its weight, which was relevant at time $t$-1 but no longer holds true at time $t$.

The Input Gate allows the cell to store novel information at time t, which may have been absent or less significant (with a low weight) at time $t$-1. The output gate regulates the information to be transmitted to the next time step ($t$+1) based on both the memory $c_t$ and the activation function. The memory vector $c_t$ enables the LSTM cell to store values over arbitrary time intervals, while the three gates control the flow of information that enters and exits the cell. The LSTM calculation equations are [16]:

$$i_t = \sigma(w_i [h_{t-1}, x_t] + b_i) \quad (9)$$

$$f_t = \sigma(w_f [h_{t-1}, x_t] + b_f) \quad (10)$$

$$g_t = \tanh(w_g [h_{t-1}, x_t] + b_g) \quad (11)$$

$$o_t = \sigma(w_0 [h_{t-1}, x_t] + b_0) \quad (12)$$

$$c_t = c_{t-1} f_t + g_t i_t \quad (13)$$

$$h_t = \tanh(c_t) \times o_t \quad (14)$$

where, $\sigma$ is the sigmoid function, $w_{i,f,g,o}$ are the weights, $b_{i,f,g,o}$ are the biases and $c_t$ is the memory cell.

The proposed LSTM model for wind speed prediction is based on numerous parts, namely

▪ Splitting dataset on training data 70% and test data 30%.

▪ Normalizing the dataset by rescaling the values to the range (0-1) using the MinMaxScaler class of scikit-learn library.

▪ Defining the three arguments of the LSTM model: Samples (values), time steps and the features, the batch size is one, one sample is processed at a time, so the time-step is one-step ahead, and we have one-dimensional output since our time series is 1-$D$.

▪ Designing and fitting the model, we use 1 neuron fit for 10 epochs to make the compilation time moderate, we apply the sigmoid activation function. The ADAM algorithm and the mean squared error loss function are used to fit optimize the model.

▪ Using the test data to assess the performance of the model. And validate the model after numerous iterations using many measures of errors metric

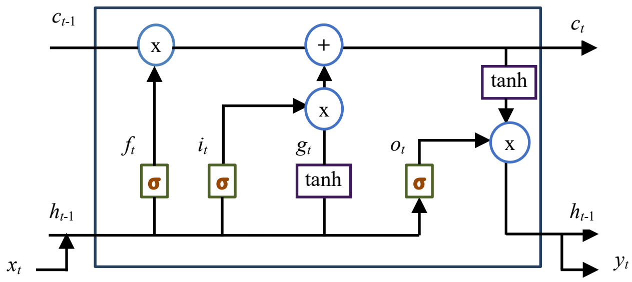



Figure 6. The LSTM cell structure

### 4.4. Bidirectional LSTM

The Bi-directional LSTM, or BiLSTM are an extension of LSTM models, the signal propagates both forward and backward in time. Firstly, the BiLSTM model supplies input data to a feedback layer; secondly, the input data sequence is reversed and supplied to the backward layer of the LSTM model. Overall, using the LSTM twice mains to improve learning long-term dependencies. Figure 7 presents the bidirectional LSTM architecture [17].

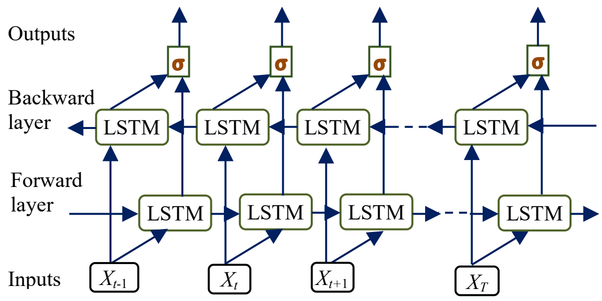



Figure 7. The BiLSTM cell structure [17]

The hidden state $H_t$ of Bi-LSTM at time $t$ comprises forward $\vec{h}_t$ and backward $\overleftarrow{h}_t$:

$$\vec{h}_t = \overrightarrow{\mathrm{LSTM}}(h_{t-1}, x_t, c_{t-1}) \quad (15)$$

$$\overleftarrow{h}_t = \overleftarrow{\mathrm{LSTM}}(h_{t+1}, x_t, c_{t+1}) \quad (16)$$

$$H_t = \left[\vec{h}_t, \overleftarrow{h}_t\right] \quad (17)$$

To build the BiLSTM model, we followed the same steps previously described in LSTM, but rather than training just one model, we introduce a pair of models. The initial model learns the input sequence, while the second model is trained on the inverse sequence of the input. Each bidirectional layer wraps a subsequent layer.

### 4.5. Gated Recurrent Unit

GRU is an enhanced version of RNN-based approaches with a structure resembling to LSTM methods, which aims to reduce the computational cost of the network. It was created by Cho, et al. [18]. The main distinctions between the architectures are: The GRU cell has no output gate; consequently, it has less parameters, the network's working memory and long-term memory are both stored in the hidden state. The single GRU cell is shown in Figure 8, $x_t$ and $s_t$, represent the actual input and output at phase $t$, while $r_t$ and $z_t$ represent respectively the reset and update gates. $z_t$ is a crucial feature, it calculates how much previous time steps are required to shift to next state. The reset gate $r_t$ decides how much prior data should be ignored. The $\hat{s}_t$ is the candidate hidden state. The sigmoid activation function is used to merge the values of the parameters.

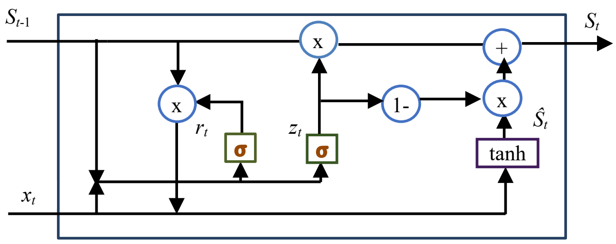



Figure 8. The GRU cell structure

The equations of the GRU cell's nodes are:

$$z_t = \sigma(w_z[s_{t-1}, x_t]) \tag{18}$$

$$r_t = \sigma(w_r[s_{t-1}, x_t]) \tag{19}$$

$$\hat{s}_t = \tanh(w[r_t s_{t-1}, x_t]) \tag{20}$$

$$s_t = (1 - z_t)s_{t-1} + z_t \hat{s}_t \tag{21}$$

where, $\sigma$ is the sigmoid function, $w_{z,r}$ are the weights.

To implement the GRU model, we followed exactly, the same steps as the LSTM model, we compile the algorithm with the same parameters and train it for the same number of epochs. The main distinction between the two models, is the number of the gates. GRU utilize less training parameters, as it has two gates (reset and update) whereas LSTM has three gates (input, output and forget).

### 4.6. Support Vector Regression

Support vector machines (SVMs) are supervised machine learning models developed to address mathematical discrimination and regression problems. In 1996, Vladimir Vapnik, et al. [19] proposed a method to use SVMs to solve regression problems, using the kernel technique, which is widely used in machine learning to solve a non-linear problem. The regression version of SVM is named Support Vector Regression (SVR). Maximum-margin classifiers or Support Vector Machines (SVMs) aim to maximize the distance between the nearest vectors of each class and the line. If the SVM attempts to split the dataset into two zones (classification), the SVR attempts to do the inverse while optimizing the epsilon distance. Figure 9 shows the basic principle of SVR.

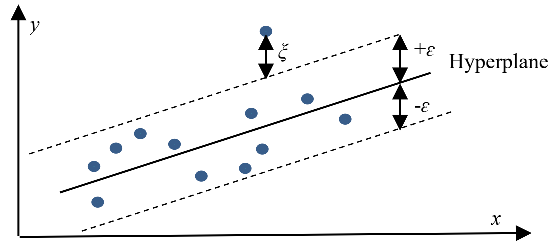



Figure 9. The basic principle of SVR

The first steps in implementing the SVR model are the same as in previous RNN algorithms, namely; separate the dataset into a train set and a test set, reshape the data to build a time-step based dataset. The aim of the regression is to fit a line (hyperplane) in feature space, in the form of a regression function that approximates the maximum number of data points with the lowest possible error. The function is described as:

$$f(x) = w^T \rho(x) + b \tag{22}$$

where, $w$ is the weight vector, $\varphi(x)$ is the mapping function, and $b$ is the bias constant. The coefficients $w$ and $b$ are estimated by minimizing the risk function.

To build the SVR model, we use the kernel function Gaussian radial basis function (RBF), RBF is counted among of the most used kernels due of its resemblance to the Gaussian distribution. For two points $x_i$ and $x_j$, the RBF kernel function calculates their similarity, The following is a mathematical representation of this kernel:

$$K(x_i - x_j) = \exp\left(-\frac{\left\|x_i - x_j\right\|^2}{2\sigma^2}\right) \tag{23}$$

where, $\sigma$ is the hyperparameter of the SVR model.

To implement the RBF Kernel SVR in the scikit-learn library of python, we set two hyperparameters $\gamma$ for the RBF Kernel and c for the SVR. $\gamma$ is inversely proportional to $\sigma$. The value of $\sigma$ is determined by using the hyperparameters tuning technique Grid Search Cross Validation. The constant c defines the amount of error allowed in the model, adjusting the limit between model complexity and error. Therefore, we fix the parameters: kernel='RBF' *gamma* = 0.5 and *C*=10.

## 5. RESULTS AND DISCUSSION

### 5.1. Prediction Accuracy per Season

Table 2 indicates the errors metrics achieved by each approach for predicting the wind speed data: ELM, CNN, LSTM, BiLSTM, GRU & SVR for each quarter of the year.

Table 2. Errors metrics values

| | | ELM | CNN | LSTM | BiLSTM | GRU | SVR |
|---|---|---|---|---|---|---|---|
| Q1 | *RMSE* | 0.322 | 0.368 | 0.299 | 0.304 | 0.309 | 0.338 |
| | *MAE* | 0.226 | 0.265 | 0.207 | 0.208 | 0.217 | 0.230 |
| | *MAPE* | 0.094 | 0.125 | 0.087 | 0.086 | 0.091 | 0.111 |
| Q2 | *RMSE* | 0.304 | 0.356 | 0.292 | 0.296 | 0.298 | 0.335 |
| | *MAE* | 0.220 | 0.278 | 0.222 | 0.244 | 0.242 | 0.250 |
| | *MAPE* | 0.091 | 0.109 | 0.088 | 0.092 | 0.093 | 0.101 |
| Q3 | *RMSE* | 0.255 | 0.325 | 0.253 | 0.249 | 0.244 | 0.306 |
| | *MAE* | 0.176 | 0.198 | 0.177 | 0.176 | 0.181 | 0.190 |
| | *MAPE* | 0.076 | 0.098 | 0.068 | 0.069 | 0.072 | 0.091 |
| Q4 | *RMSE* | 0.313 | 0.366 | 0.312 | 0.313 | 0.325 | 0.329 |
| | *MAE* | 0.230 | 0.301 | 0.218 | 0.216 | 0.221 | 0.244 |
| | *MAPE* | 0.111 | 0.136 | 0.105 | 0.104 | 0.109 | 0.130 |

The results presented in Table 2, indicate the promise forecast quality of the six-trained models in terms of the comparison analysis. As shown in Figure 10, the *MAPE* values of the proposed models are considerably lower during summer (the third quarter) than those in other seasons, with slightly lower forecasting performance in fall. These findings suggest that weather conditions significantly influence the outcome of prediction performance, as fall weather is generally more unstable and unpredictable [20]. Moreover, the average *RMSE* values confirm that RNN-based models, specifically LSTM, BiLSTM, and GRU (with values of 0.289, 0.291, and 0.294, respectively), exhibit better forecasting performance than traditional feed-forward models [21]. This can largely be attributed to RNN's ability to generate outputs using historical data, making it more appropriate for modeling sequential data compared to traditional feed-forward models that use only current data [22]. Based on errors measurements, LSTM achieves the lowest minimum *MAPE* among all models (0.068), slightly outperforming GRU and BiLSTM models. However, GRUs are less complex and thus easier to adjust.

In addition, the Extreme Learning Machine (ELM) model has shown robustness and great predictions regarding with error values very close to those of RNN approaches, outperforming SVR in terms of accuracy. It is worth noting that the predicted values of SVR are moderately stable for all seasons, with more accurate prediction when fluctuations are not significant. However, the SVR approach's limitation lies in the absence of a robust approach to learning long-term temporal dependencies [23]. Lastly, as is evident from Table 2 that the CNN models showed the worst performance with the highest *RMSE* values in each term.

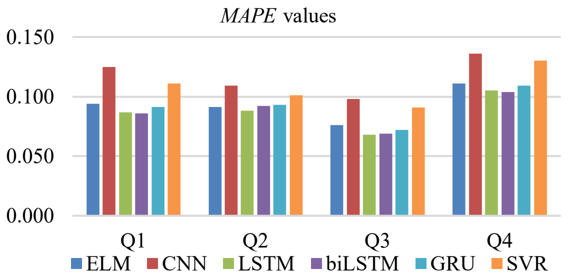



Figure 10. *MAPE* values

While, Convolutional neural networks (CNNs) are designed to handle data with grid structure, such as images, most of the research on CNNs focuses on object recognition and detection, with little discussion on prediction identification [24].

### 5.2. Prediction Accuracy per the Whole Year

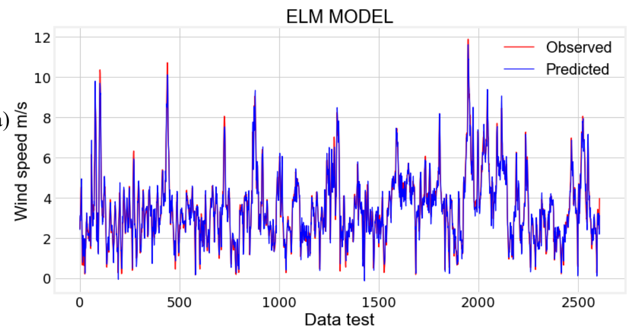
In this case, a data set of 8784 samples for the entire year was used, dividing the samples into 70% training and 30% test data. The prediction visualization graphs in Figures 11a, 11b, 11c, 11d, 11e and 11f demonstrate that all six proposed models accurately predicted the wind speed data. This suggests that these models produce reliable prediction results. Table 3 displays the *RMSE* and *MAPE* values for each method used in forecasting wind speed data for the whole year.

The RNN models once again achieved the highest accuracy score based on the evaluation metrics. The major challenge of deep learning models is the availability access to sufficient data to create a good model [21]. According to the results of this study, deep learning models provide satisfactory results with both relatively large and small data sizes. Although the statistical approach SVR performed better for smaller data sets, with an average *RMSE* of 0.327 for a quarter compared to an *RMSE* of 0.333 for the year, the ELM model still performed well, and the size of the database had negligible influence on the forecast accuracy. ELM has once again demonstrated its stability and robustness. However, even with slight improvements in performance, the CNN model was less effective compared to the other models. In summary, this study's findings suggest that RNN models are the most accurate in forecasting wind speed data. Even with small data sizes, deep learning models can still yield satisfactory results. The ELM model was also found to be stable and robust, while the CNN model was the least effective among the six proposed models.

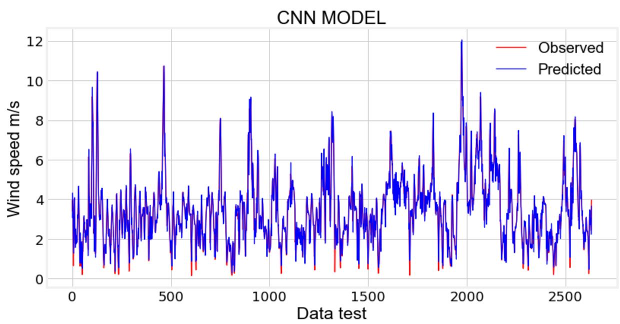
Table 3. Errors metrics for results prediction per the year

| | ELM | CNN | LSTM | biLSTM | GRU | SVR |
|---|---|---|---|---|---|---|
| *RMSE* | 0.310 | 0.349 | 0.302 | 0.310 | 0.300 | 0.333 |
| *MAPE* | 0.090 | 0.112 | 0.071 | 0.075 | 0.090 | 0.093 |

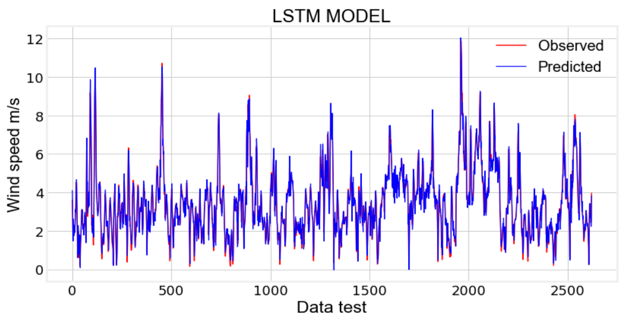
(a)




(b)




(c)



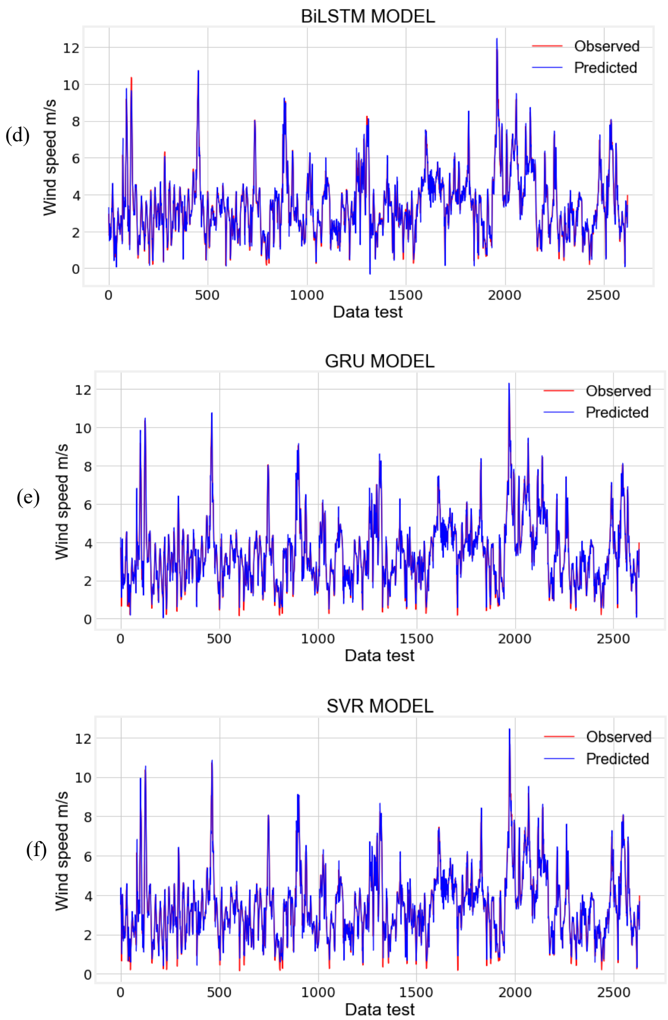



Figure 11. a) Predicted vs forecasted values, b) Predicted vs forecasted values, c) Predicted vs forecasted values, d) Predicted vs forecasted values, e) Predicted vs forecasted values, f) Predicted vs forecasted values

### 5.3. Computational Time Comparison

When implementing machine learning models, prediction accuracy is a crucial consideration. However, with a large amount of data, it is essential to consider the computational time required [25]. The computing time for the six models is presented in figure 12. The results show that the LSTM and BiLSTM models required significantly more computational time (58.2 s and 84 s) than the other machine learning models. This is expected the more complex structure which includes three weight matrices (Forget gate, Input gate and output gate), and requires more time to learn than other machine learning approaches. However, the GRU model, which only has two gates and uses fewer training parameter, has a faster training time than LSTM model [26]. In terms of execution time, the ELM and SVR models are the fastest. The ELM network determines its output weights by solving linear equations, requiring only one calculation to achieve the optimal solution. Both SVR and ELM do not require the iterative back propagation algorithm procedure which also contributes to their fast execution times.

Overall, this study found that the LSTM and BiLSTM models required more computational time than the other models due to their complex structure. Meanwhile, the ELM and SVR models were the fastest in terms of execution time. These findings highlight the importance of considering both prediction accuracy and computational time when selecting a machine learning model for a particular task.

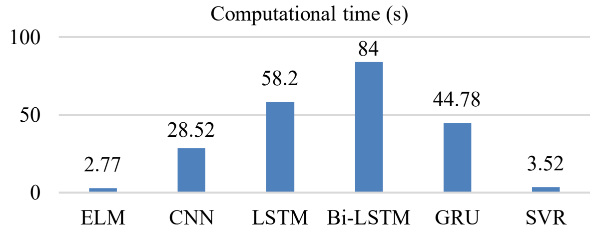



Figure 12. Computational time

### 5.4. Impact of Data Partitioning on Accuracy of Wind Speed Forecasting: Case Study ELM and LSTM

Data partitioning is crucial in enhancing the performance of machine learning models, a good generalization of the model relies on the size of the training dataset [27]. To evaluate the impact of data partitioning on prediction performance, several combinations of training and test data were applied to LSTM and ELM algorithms. Firstly, wind speed data was split into various ratios with a 5% step interval. (60%-40%, 65%-35%,70%-30%, 75%-25%, 80%-20%, 85%-15% and 90%-10%). Next, using these splitting data, different sub-datasets were generated. The model's accuracy was evaluated based on *RMSE* values, which compare the actual and the predicted values of the testing data.

This experiment aims to investigate how the performance of the LSTM and ELM algorithms is affected by the partitioning of data. By applying different data ratios, this study provides insights into the optimal training dataset size required to achieve good generalization of the model. These results have important effects for the development of machine learning models in real-world applications, where limited data may be available and accurate predictions are critical.

As shown in Figure 13, it is clearly observed that ML models are significantly influenced by the ratio used to split the datasets for training and validation. These results indicate that the selection of the amount of wind speed data for generating datasets has a considerable impact on the forecasting results. Overall, it can be confirmed that the ELM model is the most stable model under different splitting ratios The results confirmed that the training/testing ratio of 70-30%, was the most appropriate for training and validating the LSTM and ELM models, which is consistent with previous research [28], [29]. It is important to note that as the size of training data increase, the model becomes more accurate. It has been noticed that if the data training size is too large (up to 80%), a complex model can store it without comprehending the underlying relationships among the variables. This can lead to overlearning or overfitting, where the model works very well on the training data but has difficulty generalizing to new data [30]. Nevertheless, it should be noted that as the size of the training data increases, the amount of time and heap space required for each run also increases.

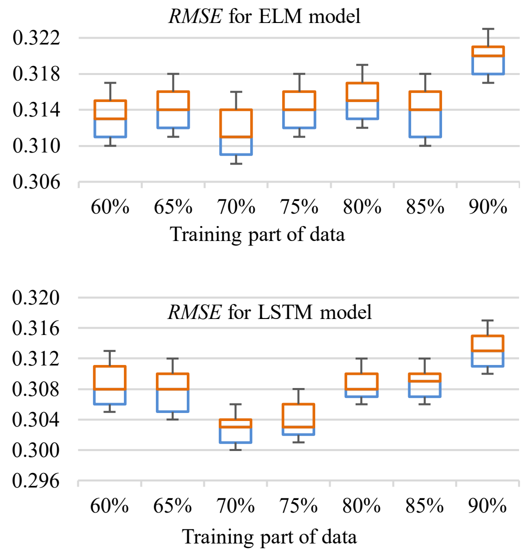



Figure 13. *RMSE* values under different splitting ratios

## 6. CONCLUSION

In conclusion, the objective of this study was to create and contrast different machine learning approaches for predicting time series of wind speed data. Six models, including SVM, CNN, RNN, LSTM, biLSTM, and ELM, were implemented and evaluated. These models were tested on each quarter of the year and on the whole year data to evaluate their performance based on different error metrics and computational time. The results of the study showed that all models performed satisfactorily, even with relatively small-sized data. The LSTM model outperformed the others in terms of the evaluation metrics dominated the other models with *MAPE* value of 7.1%, although it required more memory to train, and was sensitive to different random weight initializations. On the other hand, the GRU model was faster and more memory-efficient than LSTM and BiLSTM, making it suitable for applications where memory allocation and quick processing are important.

It is worth noting that time-series modeling and forecasting are not restricted to only recurrent models like LSTM and GRU. ELM was found to be more efficient in terms of computational time, as it reached solutions quickly than other ML techniques. Because of its quick learning capabilities from a training data set, ELM would be better suited for real-time applications where processing speed is crucial (2.77 seconds). One important consideration for selecting a split percentage is that it should meet the project's objectives while taking into account the computational cost in training and evaluating the model, as well as the representativeness of the training and test sets. Overall, this research offers valuable perspectives into the development and comparison of different machine learning approaches for predicting time series of wind speed data and sheds light on important factors that need to be considered in selecting the best approach for a given application.

## REFERENCES

[1] "bp-stats-review-2021-full-report.pdf." www.connaissancedesenergies.org/sites/default/files/pdf-actualites/bp-stats-review-2021-full-report.pdf.
[2] S.P. Kumar, "Improved Prediction of Wind Speed using Machine Learning", EAI Endorsed Transactions on Energy Web, Vol. 6, No. 23, p. 157033, June 2019.
[3] N.E. Koufi, A. Belangour, A.E. Koufi, M. Sadiq, "A Systematic Literature Review of Machine Learning Techniques Applied to Precision Marketing", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 52, Vol. 14, No. 3, pp. 185-192, September 2022.
[4] L. Xiao, W. Shao, F. Jin, Z. Wu, "A Self-Adaptive Kernel Extreme Learning Machine for Short-Term wind Speed Forecasting", Applied Soft Computing, Vol. 99, p. 106917, February 2021.
[5] R.K.B. Navas, S. Prakash, T. Sasipraba, "Artificial Neural Network Based Computing Model for Wind Speed Prediction: A Case Study of Coimbatore, Tamil Nadu, India", Physica A: Statistical Mechanics and its Applications, Vol. 542, p. 123383, March 2020.
[6] A. Khosravi, R.N.N. Koury, L. Machado, J.J.G. Pabon, "Prediction of Wind Speed and Wind Direction Using Artificial Neural Network, Support Vector Regression and Adaptive Neuro-Fuzzy Inference System", Sustainable Energy Technologies and Assessments, Vol. 25, pp. 146-160, February 2018.
[7] G. Santamaria Bonfil, "Wind Speed Forecasting for Wind Farms: A Method Based on Support Vector Regression", Renewable Energy, p. 20, 2016.
[8] V. Prema, S. Sarkar, K.U. Rao, A. Umesh, "LSTM Based Deep Learning Model for Accurate Wind Speed Prediction", ICTACT Journal on Data Science and Machine Learning, Issue 1, Vol. 1, No. 1, p. 6, December 2019.
[9] H. Liu, X. Mi, Y. Li, "Smart Deep Learning-Based Wind Speed Prediction Model Using Wavelet Packet Decomposition, Convolutional Neural Network and Convolutional Long Short-Term Memory Network", Energy Conversion and Management, Vol. 166, pp. 120-131, June 2018.
[10] C. Li, G. Tang, X. Xue, A. Saeed, X. Hu, "Short-Term Wind Speed Interval Prediction Based on Ensemble GRU Model", IEEE Trans. Sustain. Energy, Vol. 11, No. 3, pp. 1370-1380, July 2020.
[11] J. Chen, G.Q. Zeng, W. Zhou, W. Du, K.D. Lu, "Wind Speed Forecasting Using Nonlinear-Learning Ensemble of Deep Learning Time Series Prediction and Extremal Optimization", Energy Conversion and Management, Vol. 165, pp. 681-695, June 2018.
[12] J. Song, J. Wang, H. Lu, "A Novel Combined Model Based on Advanced Optimization Algorithm for Short-Term Wind Speed Forecasting", Applied Energy, Vol. 215, pp. 643-658, April 2018.
[13] W. Deng, B. Ye, J. Bao, G. Huang, J. Wu, "Classification and Quantitative Evaluation of Eddy

Current Based on Kernel-PCA and ELM for Defects in Metal Component", Metals, Vol. 9, p. 155, February 2019.
[14] F.K. Al Jibory, O.A. Mohammed, M.S.H. Al Tamimi, "Age Estimation Utilizing Deep Learning Convolutional Neural Network", International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 53, Vol. 14, No. 4, pp. 219-224, December 2022.
[15] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, D.J. Inman, "1D Convolutional Neural Networks and Applications: A Survey", Mechanical Systems and Signal Processing, Vol. 151, p. 107398, April 2021.
[16] I. Tyass, T. Khalili, R. Mohamed, B. Abdelouahad, A. Raihani, K. Mansouri, "Wind Speed Prediction Based on Statistical and Deep Learning Models", International Journal of Renewable Energy Development, Vol. 12, pp. 288-299, January 2023.
[17] S. Siami Namini, N. Tavakoli, A.S. Namin, "The Performance of LSTM and BiLSTM in Forecasting Time Series", The IEEE International Conference on Big Data, pp. 3285-3292, Los Angeles, CA, USA, December 2019.
[18] K. Cho, B. van Merrienboer, D. Bahdanau, Y. Bengio, "On the Properties of Neural Machine Translation: Encoder-Decoder Approaches", The Eighth Workshop on Syntax (SSST-8), Semantics and Structure in Statistical Translation, Doha, Qatar: Association for Computational Linguistics, pp. 103-111, October 2014.
[19] C. Cortes, V. Vapnik, "Support-Vector Networks", Mach Learn, Vol. 20, No. 3, pp. 273-297, September 1995.
[20] P. Kou, C. Wang, D. Liang, S. Cheng, L. Gao, "Deep Learning Approach for Wind Speed Forecasts at Turbine Locations in a Wind Farm", IET Renewable Power Generation, Vol. 14, No. 13, pp. 2416-2428, 2020.
[21] A. Dairi, F. Harrou, A. Zeroual, M.M. Hittawe, Y. Sun, "Comparative Study of Machine Learning Methods for COVID-19 Transmission Forecasting", Journal of Biomedical Informatics, Vol. 118, p. 103791, June 2021,
[22] Q. Tao, F. Liu, D. Sidorov, "Recurrent Neural Networks Application to Forecasting with Two Cases: Load and Pollution", Intelligent Computing and Optimization, P. Vasant, I. Zelinka, G.W. Weber, (Eds.), Advances in Intelligent Systems and Computing, Vol. 1072. Cham: Springer International Publishing, pp. 369-378, 2020.
[23] X. Zhou, Z. Liu, F. Wang, Y. Xie, X. Zhang, "Using Deep Learning to Forecast Maritime Vessel Flows", Sensors, Vol. 20, No. 6, Art. No. 6, January 2020.
[24] S. Kareem, Z.J. Hamad, S. Askar, "An Evaluation of CNN and ANN in Prediction Weather Forecasting: A Review," Sustainable Engineering and Innovation, Vol. 3, No. 2, Art. No. 2, October 2021.
[25] J. Fan, et al., "Empirical and Machine Learning Models for Predicting Daily Global Solar Radiation from Sunshine Duration: A Review and Case Study in China", Renewable and Sustainable Energy Reviews, Vol. 100, pp. 186-212, February 2019.
[26] R.A. Rajagukguk, R.A.A. Ramadhan, H.J. Lee, "A Review on Deep Learning Models for Forecasting Time Series Data of Solar Irradiance and Photovoltaic Power", Energies, Vol. 13, No. 24, Art. No. 24, January 2020.
[27] R. Medar, V. Rajpurohit, B. Rashmi, "Impact of Training and Testing Data Splits on Accuracy of Time Series Forecasting in Machine Learning", International Conference on Computing, Communication, Control and Automation (ICCUBEA), pp. 1-6, August 2017.
[28] B.T. Pham, I. Prakash, A. Jaafari, D.T. Bui, "Spatial Prediction of Rainfall-Induced Landslides Using Aggregating One-Dependence Estimators Classifier", J Indian Soc Remote Sens, Vol. 46, No. 9, pp. 1457-1470, September 2018.
[29] Q.H. Nguyen, et al., "Influence of Data Splitting on Performance of Machine Learning Models in Prediction of Shear Strength of Soil", Mathematical Problems in Engineering, Vol. 9, p. e4832864, February 2021.
[30] P. Domingos, "A Few Useful Things to Know About Machine Learning", Commun. ACM, Vol. 55, No. 10, pp. 78-87, October 2012.

## BIOGRAPHIES

Name: **Ilham**
Surname: **Tyass**
Birthday: 07.12.1980
Birthplace: Mohammedia, Morocco
Master: Engineering, Electro-Mechanics Department, ENSMR School, Rabat, Morocco, 2004
Doctorate: Student, ENSET Institute, Hassan II University, Casablanca, Mohammedia, Morocco, Since 2018
Research Interests: Optimization and Numerical Modeling, Hybrid Wind-Photovoltaic System, Energy Storage
Scientific Publications: 3 Papers

Name: **Abdelhadi**
Surname: **Raihan**i
Birthday: 01.01.1968
Birthplace: El Jadida, Morocco
Bachelor: Electronics Systems, Electrical Engineering, ENSET, Hassan II University, Casablanca, Mohammedia, Morocco, 1991
Master: Information Processing, Ben M'sik University, Casablanca, Morocco, 1994
Doctorate: Parallel Architectures Application and Image Processing, Ain Chock University, Casablanca, Morocco, 1998
The Last Scientific Position: Prof., Department of Electrical Engineering, ENSET of Mohammedia, Hassan II University, Casablanca, Morocco, Since 1992
Research Interests: Multiple Domains of Medical Image Processing and Electrical Engineering, Energy Management Systems, Power and Energy Systems Control, Smart Grids
Scientific Publications: 134 Papers, 15 Theses
Scientific Memberships: IRESEN under the Grant "Green INNO Project / UPISREE"

Name: **Khalifa**
Surname: **Mansouri**
Birthday: 20.05.1968
Birthplace: Azilal, Morocco
Bachelor: Mechanical Construction, ENSET, Hassan II University, Mohammedia, Morocco, 1991
Master: Structural Calculation, Faculty of Sciences, Mohamed V University, Rabat, Morocco, 1992
Doctorate: Computer-Aided Structural Design and CAD, Faculty of Sciences, Mohamed V University, Rabat, Morocco, Inria Recqencourt, France, 1994 - Computer Science, Faculty of Science and Technology of Mohammedia, Hassan II University, Casablanca, Morocco, 2016
The Last Scientific Position: Prof., Computer Science Since 1991, Deputy Director in Charge of Training and Director, M2S2I Research Laboratory, ENSET, Hassan II University, Mohammedia, Morocco, Since 2021
Research Interests: Information Systems, E-Learning Systems, Real Time Systems, Industrial Systems (Modeling, Optimization, Numerical Computation), Artificial Intelligence
Scientific Publications: 363 Papers, 10 Books, 30 Theses