Optimally designed Variational Autoencoders for Efficient Wind Characteristics Modelling

Srinivas Soumitri Miriyala
Department of Chemical Engineering
Indian Institute of Technology, Hyderabad
Hyderabad, India
ch13m15p000002@iith.ac.in

Subhankar Chowdhury
Department of Electrical Engineering
Indian Institute of Technology, Hyderabad
Hyderabad, India
ee18mtech11035@iith.ac.in

NagaSree Keerthi Pujari
Department of Chemical Engineering
Indian Institute of Technology, Hyderabad
Hyderabad, India
ch18mtech11008@iith.ac.in

Kishalay Mitra
Department of Chemical Engineering
Indian Institute of Technology, Hyderabad
Hyderabad, India
kishalay@che.iith.ac.in

***Abstract*—Wind energy is increasingly applied as a large scale clean energy generating alternative to fossil fuels. However, limited amount of real wind data results in inaccurate construction of Wind Frequency Maps (WFMs), which model the stochastic nature of wind. The inaccuracies in WFMs may lead to over or under estimation of wind power eventually causing significant losses to wind-farmers. Hence, to resolve this crisis, deep generative models such as convolutional Variational Autoencoders (VAEs) are implemented in this work to enable accurate construction of WFMs from limited amount of real wind characteristics data. However, the heuristics based estimation of hyper-parameters in VAEs decrease their efficiency. Thus, in this work, a novel multi-objective evolutionary neural architecture search (NAS) strategy is devised for simultaneously estimating the optimal number of convolutional and feedforward layers, number of filters/nodes in each layer, filter size, pooling option and nonlinear activation choice in VAEs. The proposed framework is designed to balance the conflicting objectives of generalizability and parsimony in VAEs, thereby reducing the chances of their over-fitting. The optimally designed VAE (with 92% accuracy) is used to generate new wind frequency scenarios for accurate construction of WFM. Additionally, the effect of number of new scenarios required for accurate WFM construction is also studied while performing the comparison with an ideal case. It was found that WFM constructed with original limited data resulted in 9% deficit in energy calculation from a single wind turbine, justifying the need for generative models such as VAEs for accurate wind characteristics modelling.**

***Keywords*— Deep Generative modelling, Neural Architecture Search, multi-objective optimization, evolutionary algorithm, Variational Autoencoder, Wind energy conversion systems, Wind frequency maps.**

I. INTRODUCTION

Global energy crisis and increasing levels of greenhouse gas emissions have driven the industries towards harnessing renewable energy sources such as solar, wind, geothermal and tidal energy. Though, the renewable resources are unlimited in support, the methods to extract energy efficiently from these sources are still at nascence. Among all the renewable energy sources, wind energy turns out to be one of the promising alternative to fossil based fuel sources due to its cleaner energy production and easy availability. Significant amount of research in wind energy conversion systems over the past decade have given it a decent shape, leading to an increase in total installed generation capacity across the world. Wind energy is harnessed by installing wind turbines, where the power produced from the turbine heavily depends on wind speed and wind direction [1]. However, the uncertain nature of wind adds difficulty in harvesting power from a wind farm leading to huge variability in power production. Therefore, studies like wind farm layout optimization and wind farm control depend heavily on mathematical models, which emulate the wind behavior accurately [1]. One type of such probabilistic model represents the wind as a joint Probability Mass Function (PMF) of wind speed and wind direction. This distribution, also known as Wind Frequency Map (WFM), is used extensively in all the research pertaining to wind energy conversion systems.

The construction of WFMs depends significantly on the time series data of wind speed and direction. However, the scarcity in the amount of wind characteristics data, poor maintenance of existing data and high labor cost associated with collecting such data have become hurdles in the way for constructing accurate WFMs. These inaccurate WFMs result in unrealistic power production through wind farm simulations resulting in significant mismatch with real power production [2]. In this regard, the accurate modelling of the WFMs can be of great help to the wind farmers for realistic power production.

In the last few decades, an extensive research on accurate forecasting of wind characteristics has been performed using different methods. Wind, being sequential in nature, many of the papers have proposed methods from conventional time series modelling and signal decomposition to model wind characteristics data [3 and 4]. The difficulties such as complex weather conditions, lack of maintenance in physical models and extreme nonlinearity in the data brought enormous scope for deep neural networks for modelling wind characteristics. A recent literature survey concluded that convolutional neural networks with k-means clustering algorithm forecast the data accurately when compared to other approaches [4]. In [5], the authors proposed a multi-modal short-term wind speed prediction framework based on denoising and prediction modules. In denoising module, the optimal configuration of stacked denoising auto encoders have been used and are applied in training the data to reduce the noise present in the data. The sinusoidal rough neural networks are used in prediction module to forecast the wind speed. And both these modules are stacked together to comprise into deep structure network to predict the wind speed data accurately. In [6], the authors proposed a deep

Authors would like to acknowledge the support provided by the SPARC project (SPARC/2018-2019/P1084/SL) funded by the Ministry of Human Resources Development (MHRD), Government of India, for this work.

learning model based on auto-encoder to predict the energy demand accurately. In [7], the authors proposed a hybrid approach using auto-encoder and Long Short Term Memory networks (LSTMs) for accurate power forecasting. The authors used different combinations of Deep Learning and Artificial Neural Network algorithms, such as Deep Belief Networks, autoencoder, and LSTMs to show the forecasting strength compared to a standard multi-Layer Perceptron networks (MLPs) and physical models. The results have shown that deep learning algorithms are superior to other techniques. A two-staged auto encoder is designed and incorporated the structural features into prediction framework for better results in wind power prediction [8]. In the contemporary times, the auto-encoders acquired the attention from researchers in the field of forecasting. The researchers are trying to incorporate the auto-encoders along with the deep learning algorithms in order to increase the prediction accuracy. In [9], the authors combined convolutional autoencoder (CAE) and LSTMs to predict time series data with high noise. The data features are extracted using one dimensional convolution for encoding and decoding the network. Later, the LSTMs are used to predict the data accurately. The results have shown that the CAE-LSTM model predicted the data accurately compared to other techniques.

Most of the articles in literature have considered modelling wind characteristics using sequential methods, such as time series modelling. The idea in these methods is to enable forecasts of wind characteristics data and then utilize this data either directly in control and optimization studies of wind energy conversion systems or for accurate construction of WFMs. However, the latter application has seen less success because, to build accurate WFMs, significantly large amount of forecast is necessary. Even the state-of-the-art forecasting methods are known to be accurate only for short-term forecasts. Therefore, in this work, an attempt is made to use the generative modelling aspect to construct accurate WFM with existing limited amount of data. The time series data of wind speed and direction is converted into a series of WFMs. Each of these WFMs are then treated as images and generative model such as the Variational Autoencoder (VAE) is trained on this data to generate new WFMs. The real WFMs are then combined with generated WFMs to create a single accurate WFM.

VAEs are versatile tools in deep learning which have recently gained immense applicability across various domains as robust generative models. For instance, in [10], the authors used VAEs to forecast the futuristic events from static images. In another recent work, the VAEs are used for forecasting the road traffic systems [11]. In [12] the authors proposed variational-LSTM autoencoder approach to model the spread of COVID-19 for each country across the globe. Despite of immense popularity, the difficulty with VAEs lies in the selection of associated hyper-parameters – architecture of VAE, filter size, pooling option and activation function. Hence, the optimal design of VAEs to model the wind frequency maps is the need of hour.

The growth and applicability of deep learning in various fields has led to significant amount of research in a novel area called automated machine learning where the models are developed to be capable of smart and automated estimation of hyper-parameters [13]. Neural Architecture Search (NAS) is one such topic which aims at eliminating the heuristics based approach for determining the architecture of ANNs [14]. Broadly, the NAS strategies are classified into three categories: a) NAS based on reinforcement learning [15], b) NAS based on unsupervised learning [16] and c) optimization formulation based NAS [17]. In different works of literature, VAEs are used extensively across many domains, including the field of wind speed forecasting. However, to the best of our knowledge, no work has been reported where wind characteristics (speed and direction) are modelled jointly as a probability mass function, then converted into greyscale images and predicted using optimally designed VAEs. The highlights of this work are described as follows:

- A multi-objective evolutionary algorithm is used for optimally designing VAEs, while balancing the aspects of accuracy and parsimony.
- The proposed optimization framework is solved using evolutionary optimization solver, non-dominated genetic algorithm, NSGA-II [18].
- A set of Pareto solutions are obtained as a result of proposed algorithm, where each solution is a potential VAE architecture, which is as significant as other solutions in the Pareto list.
- A single VAE model is obtained from the Pareto list by implementing the held-out sample test thereby ensuring that the least over-fitted model is selected from the list of solutions.
- The real wind characteristics time series data is divided into daily Wind Frequency Maps. Each WFM is transformed into a greyscale image, which are then collectively used for training and validating the VAE.
- Optimally designed VAE after getting trained with the given WFMs, generates new WFMs, which are then combined to form a single accurate WFM.
- Using the accurate WFM and a wind speed and direction signal for one year, wind power generated from a single turbine is tabulated for comparison with power generated from same turbine using WFM from original data.
- The proposed NAS strategy for VAE is generic and can be used irrespective of size and type of data generated from any domain.
- All the simulations are conducted using codes generated in Fortran 90 environment and no open source is used for optimization or deep learning tools used in proposed work.

In the rest of the paper, Section II presents real wind characteristics data. Then the procedure for converting this data into WFMs and the images for training VAE is presented followed by a brief description of VAE and the proposed novel algorithm of optimal design of VAE networks. Section III describes the results of present work and comparative study. Section IV summarizes the conclusions.

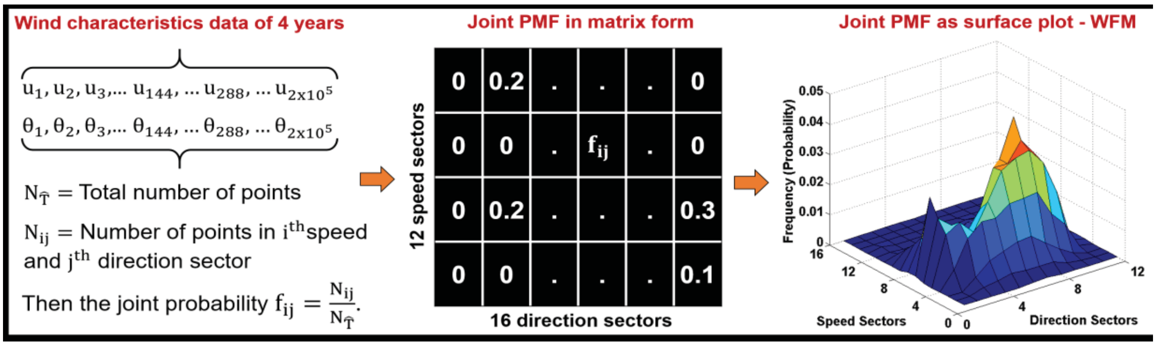

Figure 1. Construction of wind frequency map from wind characteristics data collected over 4-years.

II. FORMULATION

A. Data Description

Four-year wind speed and direction data are obtained from a French electric utility company called ENGIE [19]. The data was collected from a wind farm in La Haute Borne, France using anemometers. In order to remove the noise, the data was pre-processed using moving average method. The procedure for construction of WFM from such a time series data is described below and pictorially represented in Fig. 1.

a) Wind speed data is divided into several (e.g. 16) speed bins such that each bin has an equal range of wind

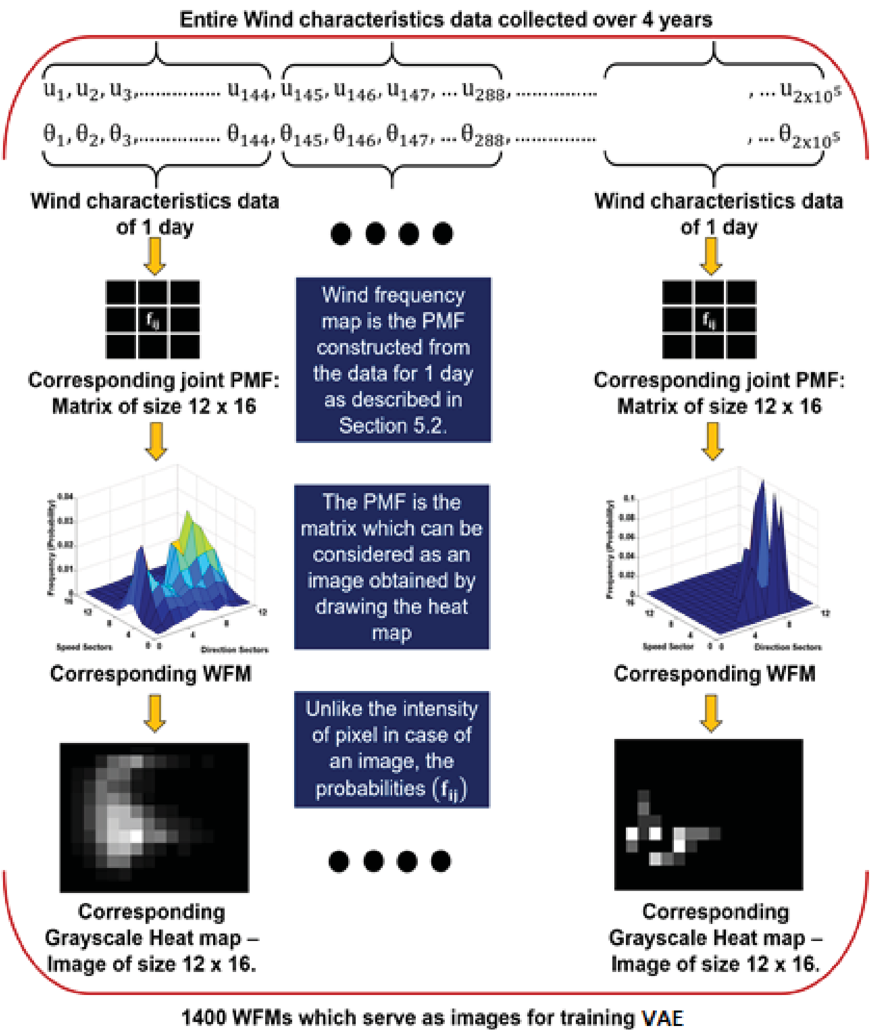

Figure 2. Generation of the image database from wind characteristics' times series data, for training VAE.

speed. Similarly, the direction data is divided into 12 equal sectors.

b) From the time series data, the number of data points in i^{th} speed bin and j^{th} direction sector are counted and denoted by n_{ij}.

c) The frequency f_{ij} is calculated by dividing n_{ij} with total number of data points.

The joint PMF constructed in this format can be visualized as a 12 x 16 matrix, where each element of the matrix is defined by the probability value f_{ij} . The normalized heat-map corresponding to this matrix is 2 dimensional greyscale image, where each probability value f_{ij} can be considered as a pixel value. Such an image whose size will be 12 x 16 is used a sample point for training the VAE. The procedure for constructing the training data is illustrated in Fig. 2, where a total of $2x10^6$ data points (collected at 10-minute resolution over four years) are divided into 1400 WFMs (each WFM corresponding to a single day's data).

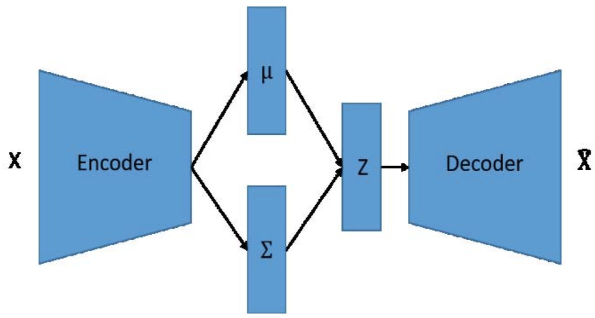

Figure 3. Architecture of a Variational Autoencoder

B. *Variational Autoencoder*

VAEs belong to the class of neural networks called autoencoders which were designed to compress high dimensional data into low dimensional representation called latent vectors [20]. The autoencoders are made up of two parts – encoder, which would transform the input image into the latent vector and the decoder, which would transform the latent vector into the image. The autoencoders are trained to mimic the input data provided to them. Thus, unlike the conventional networks which need labelled data, the autoencoders work with data where the inputs and outputs are same. Due to this reason, the autoencoders do not have the capability of generating new data, i.e., they are trained to only compress the given data. However, the VAEs are designed such that, the latent space generated by them is continuous, thus allowing for easy random sampling and interpolation [21]. In VAEs, the encoder outputs two vectors instead of a single latent vector as is the case in autoencoders. One of these two vectors is the vector of means while the other is the vector of standard deviations. They form the parameters of a multi-dimensional Gaussian distribution governing the random latent variables. Using this distribution, a latent vector is randomly sampled and sent as input to the decoder which will generate the network output. This procedure is depicted in Fig. 3. Once the VAE is trained, one can select only the decoder, draw the random samples from the multi-variate Gaussian distribution whose parameters were learnt during the process of training and send them as inputs to decoder to generate new images. This procedure brings in stochastic nature in the otherwise deterministic autoencoders, thus making them probabilistic generative models. Honoring the space constraints, the rigorous mathematical treatment of VAEs is not discussed in the current manuscript. Interested readers can refer [20 and 21] for more details on training algorithm of VAEs. In order to model the image based dataset, both encoder and decoder in the VAE used in this work are convolutional neural networks [20], whose hyper-parameter estimation algorithm is described next.

C. *Algortihm for optimal design of VAEs*

The predictability of any model can be improved by increasing the parameters in the model. However, the issue of over-fitting arises with a drastic increase in number of parameters. Several methods are described in the literature to overcome the issue of over-fitting, however, most of these methods are heuristic in nature. Hence the authors proposed an optimization perspective of solving the issue of over-fitting. A novel two objective optimization framework is formulated as shown in Eq. 1. To address the trade-off between number of parameters in the model and accuracy, the objectives are considered as minimizing the Root Mean Square Error (RMSE) measured on the test set along with total parameters in the VAE network. A symmetric VAE is implemented in this work, i.e., the architecture configuration of encoder and decoder are mirror images of each other. Thus, in the hyper-parameter estimation algorithm, only the hyper-parameters in either decoder or encoder need to be estimated. The hyper-parameters described below, serve as the decision variables in the optimization formulation presented in Eq. 1.

i. n^K - a binary variable for filter (kernel) size during convolution: Two choices for kernel size are explored. If $n^K = 1$, kernel is 3×3 and if $n^K = 2$, kernel is 5×5.

ii. n^P - a binary variable for pooling. Due to the dimensions of the WFM being much smaller than those of conventional images, n^P indicates whether to consider ($n^P = 1$) the pooling or not ($n^P = 2$). If the pooling is considered, then average pooling with receptive field of size 2 is implemented.

iii. $\{n_l^C\}$ – set of variables to indicate number of filters in convolutional layers. The size of this set is maximum number of convolutional layers explored. $n_l^{C^{LB}}$ and $n_l^{C^{UB}}$ are lower and upper bounds on each variable in this set.

iv. $\{n_l\}$ – set of variables to indicate number of nodes in fully connected layers. The size of this set is maximum number of fully connected layers explored. n_l^{LB} and n_l^{UB} are lower and upper bounds on each variable in this set.

v. n_{LV} – dimensions of the latent vector.

vi. n^{AF} – a binary variable to indicate the activation function in CAE. If $n^{AF} = 1$, activation is tan-sigmoid and if $n^{AF} = 2$, activation is rectified linear unit (ReLU) [20].

$$\min_{n^K,\ n^P,\ \{n_l^C\},\ \{n_l\},\ n_{LV}\text{ and }n^{AF}} (\text{Accuracy in terms of RMSE, Total parameters } \overline{P}) \quad (1)$$

where,

$$\text{RMSE} = \frac{1}{\overline{N}}\sum_{n=1}^{\overline{N}}\left(\sqrt{\frac{1}{192}\sum_{i=1}^{12}\sum_{j=1}^{16}\left(Y_{ij}^{\text{Image}} - \hat{Y}_{ij}^{\text{Image}}\right)^2}\right)$$

is evaluated on test set of size $\overline{N}$ where Y_{ij}^{Image} and $\hat{Y}_{ij}^{\text{Image}}$ are original and network generated pixel values in the image

subject to,

$${n^C}_l^{LB} \le n_l^C \le {n^C}_l^{UB}\ \forall\, l = 1 \text{ to } L^C = 3$$

$$n_l^{LB} \le n_l \le n_l^{UB}\ \forall\, l = 1 \text{ to } L^{MLP} = 3$$

$$n_{LV}^{LB} \le n_{LV} \le n_{LV}^{UB}$$

$${n^C}_l^{LB} = \begin{Bmatrix}1, \text{if } l = L^C\\ 0, \text{if } l < L^C\end{Bmatrix}, n_l^C, \in \mathbb{Z}_+$$

$$n_l^{LB} = \begin{Bmatrix}1, \text{if } l = L^{MLP}\\ 0, \text{if } l < L^{MLP}\end{Bmatrix}, n_l, \in \mathbb{Z}_+$$

$$n^K,\ n^P,\ n^{AF} \in \{1, 2\}$$

Since the network is symmetric, L^{MLP} and L^C indicate the number of fully connected and convolutional layers to be explored in either encoder or decoder. To ensure that at least one convolution layer is present, the lower bound on number of nodes in $(L^C)^{th}$ layer is set as 1, i.e. ${n^C}_{L^C}^{LB} = 1$, while for other convolutional layers, it is set as 0. Similarly, for fully connected portion of VAE, the same thing is ensured by setting the lower bound on number of nodes in $(L^{MLP})^{th}$ layer as 1 and for others, the lower bound = 0.

The proposed multi-objective formulation is solved using NSGA-II. When NSGA-II gives the decision variable set, first the architecture of VAE is determined. For instance, if the decision variable set given by NSGA-II is [1, 2, {3, 0, 5}, {2, 4, 2}, 5, 1], following the sequence n^K, n^P, $\{n_l^C\}$, $\{n_l\}$, n_{LV} and n^{AF}, as mentioned in Eq. 1, then {3, 0, 5} corresponds to nodes in 1st to 3rd convolutional layer and {2, 4, 2} corresponds to nodes in 1st to 3rd fully connected (FC) layers. However, number of nodes in second convolutional layer is 0, indicating that the configuration of VAE will be consisting of 2 convolutional layers, 7 FC layers and 2 de-convolutional layers: 3-5-input-2-4-2-5-2-4-2-output-3-1, where input and output correspond to length of unraveled vector. The size of latent feature vector in this VAE is 5. Further, since the first value of the given decision variable set is 1, a kernel of size 3×3 is used for convolution and deconvolution. Since the second decision variable is 2, it means that pooling will not be implemented and since the last decision variable is 1, it means that activation is by tan-sigmoid function across the entire network. In another example, let the decision variable set be [2, 2, {4, 10, 6}, {3, 0, 8}, 7, 2]. Then the VAE configuration will be 4-10-6-input-3-8-7-8-3-output-10-4-1, indicating, 3 convolutional layers with 4, 10 and 6 nodes, respectively, followed by 5 fully connected layers with 3, 8, 7, 3 and 8 nodes, respectively and 3 de-convolutional layers containing 10, 4 and 1 nodes, respectively. The length of the latent vectors for this example is 7, kernel size is 5×5 with no pooling and ReLU activation is applied across the network. In this way, once the VAE is designed, the number of parameters (weights and biases in the entire network) is determined (2nd objective function). The VAE network is then trained using Adam optimization method [20].

After training, the network is tested with unseen images of WFMs to evaluate the 1st objective (RMSE), which along with the 2nd objective is sent back to NSGA-II after storing them in the database, which is maintained to prevent redundancy. This completes the loop of single function evaluation in NSGA-II and this loop repeats for all populations in the generation. The crossover, mutation and selection operations of NSGA-II are implemented to create a new generation [18] and the procedure is repeated till termination. Once the algorithm terminates, it will result in a set of equally significant solutions called the Pareto solutions. To select a single architecture from this list, a held-out test is conducted. All the Pareto solutions (which are different VAE architectures) are subjected to predict for a completely unseen set of images (held-out set) and the model whose accuracy is maximum is considered as the final solution of the proposed algorithm.

D. WFM generation and power calculation

Once the optimal VAE is designed, it will be used to generate new WFMs. However, since the VAE was trained with daily WFM scenarios, the newly generated WFMs will also be daily scenarios. To perform further analysis (power calculation) these WFMs (given data and generated data) must be combined into single WFM. The procedure for constructing a single frequency map from a set of maps generated from VAE is illustrated in following steps.

1. Any element in the frequency map is given by $f_{ij} = \frac{N_{ij}}{N_{Total}}$ where N_{Total} is the total number of points in the time series data and N_{ij} is the number of points in i^{th} bin of speed and j^{th} sector of wind direction.
2. Using the value of F_{ij} and N_{Total}, N_{ij} can be obtained $\forall$ i and j in all the frequency maps (original and generated).
3. A single frequency map can be then obtained by combing M frequency maps using $F_{ij}^{combined} = \frac{\sum_{k=1}^{M} N_{ij}^k}{N_{Total} * M}$ $\forall$ i and j in the combined map.

In this work, the effect of number of frequency maps used for constructing a single accurate WFM is also studied. While 1400 maps were used to train and test the VAE model, 2000, 5000 and 10000 new WFMs were generated, i.e., the following cases are considered: M = 1400+2000, 1400+5000 and

1400+10000, along with the WFM built using the original 1400 data. By applying the law of large numbers, the case with M = 1400+10000 is considered as the ideal case for comparison. The expected power from a single turbine is calculated using the formula in Eq. 2. The exact power produced from the turbine, $P_{Curve}(u_r)$, is a function of wind speed, which is provided by the turbine manufacturer [2].

$$P_{turbine} = P_{curve}(u_r) \times WFM(u_r, \theta_d)$$

$$\text{Expected Power} = \sum_{d=1}^{12} \sum_{s=1}^{u_{max}=16} P_{turbine} \quad (2)$$

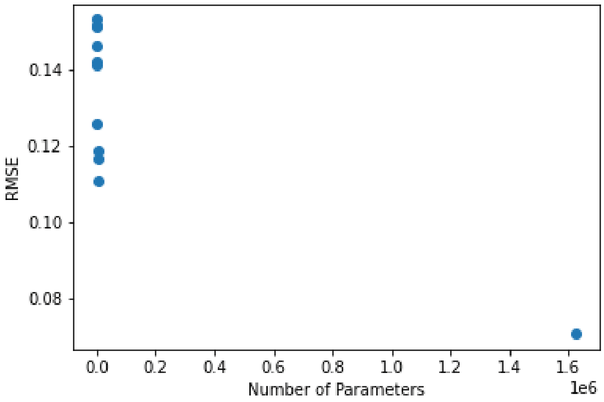

Figure 4. Pareto front obtained by solving the two objective optimization problem (minimize RMSE and minimize the total number of parameters in the VAE) for optimal design of VAEs using NSGA-II.

III. RESULTS AND DISCUSSIONS

First the collected wind characteristics (speed and direction) time series data is transformed in 1400 WFMs resembling the case for each day. Out of these 1400 WFMs, 70% data was used for training and the rest were used for testing and validation. With this dataset, the proposed algorithm for optimal design of VAE is run with 100 populations and 100 generations in NSGA-II. In this work, a maximum of three convolutional and three FC layers were explored. Fig. 4 presents 2 dimensional Pareto front obtained from the proposed algorithm. NSGA-II was run for sufficiently large number of generations beyond 100 (~500) to ensure convergence of Pareto front. Each point in the Pareto front is a VAE model as shown in Table 1 along with the results of held-out test. The row in bold indicate the selected architecture. The emergence of multi-layered deep architectures against the common belief of a single layered architectures for convolution and fully connected networks justifies the need for the neural architecture search algorithm. The densely populated and widely spread Pareto front shown in Fig. 4 speaks about the importance of the NAS algorithm being multi-objective and evolutionary in nature. Among the list of Pareto solutions, the model with minimum held-out RMSE is considered to be least over-fitted. The selected model is a VAE with 2 convolutional layers containing 8 and 12 filters, respectively, followed by 3 FC layers with 228, 53 and 40 nodes, respectively, 5 dimensional latent vectors. This means the architecture of VAE is: [8-12-I-228-53-40-5-40-53-228-I-8-1] where I, the length of unraveled vector is 1152. Total number of parameters in this network is 16,23,196 and it emulates the training and test data with an RMSE of 0.08 and 0.085, respectively. The filter size and activation function is found to be 3x3 and ReLU, respectively and pooling layer is not present in this network. Out of WFM images used for testing, the performance of optimal VAE is shown for few selected images in Fig. 5. Similar results were obtained for all other images in test set and training set, however, they are not shown here due to the space constraints.

Using the optimal VAE model, new WFMs are generated for M = 1400+2000, 1400+5000 and 1400+10000 as explain in Section II. The frequency maps corresponding to these cases along with the heat-map images are shown in Fig. 6.

Table 1. List Pareto solutions obtained from the proposed algorithm for optimal design of VAEs.

S. No	$\mathbf{n^K}$	$\mathbf{n^P}$	$\mathbf{n_1^C}$	$\mathbf{n_2^C}$	$\mathbf{n_3^C}$	$\mathbf{n_1}$	$\mathbf{n_2}$	$\mathbf{n_3}$	$\mathbf{n_{LV}}$	$\mathbf{n^{AF}}$	$\mathbf{\bar{P}}$	**RMSE**	**Held out Test RMSE**
1	2	2	4	1	0	64	0	0	2	2	1958	0.160032	0.161148
2	2	2	5	4	0	64	0	0	3	2	6103	0.125807	0.165431
3	2	2	1	1	0	64	1	12	2	2	1518	0.172249	0.174429
4	2	2	2	1	0	64	0	0	2	2	1754	0.16121	0.168313
5	2	2	4	3	0	64	0	0	2	2	4424	0.134605	0.138028
6	2	2	1	1	0	64	1	4	3	2	1452	0.173734	0.183013
7	2	2	5	4	0	64	0	0	2	2	5909	0.132242	0.137702
8	2	2	1	1	0	64	0	0	2	2	1652	0.166057	0.167271
9	2	2	1	2	0	64	0	0	2	2	2735	0.142495	0.176423
10	2	2	2	1	0	64	1	7	2	2	1570	0.171545	0.171347
11	**1**	**1**	**8**	**12**	**0**	**228**	**53**	**40**	**5**	**2**	**1623196**	**0.080044**	**0.085851**

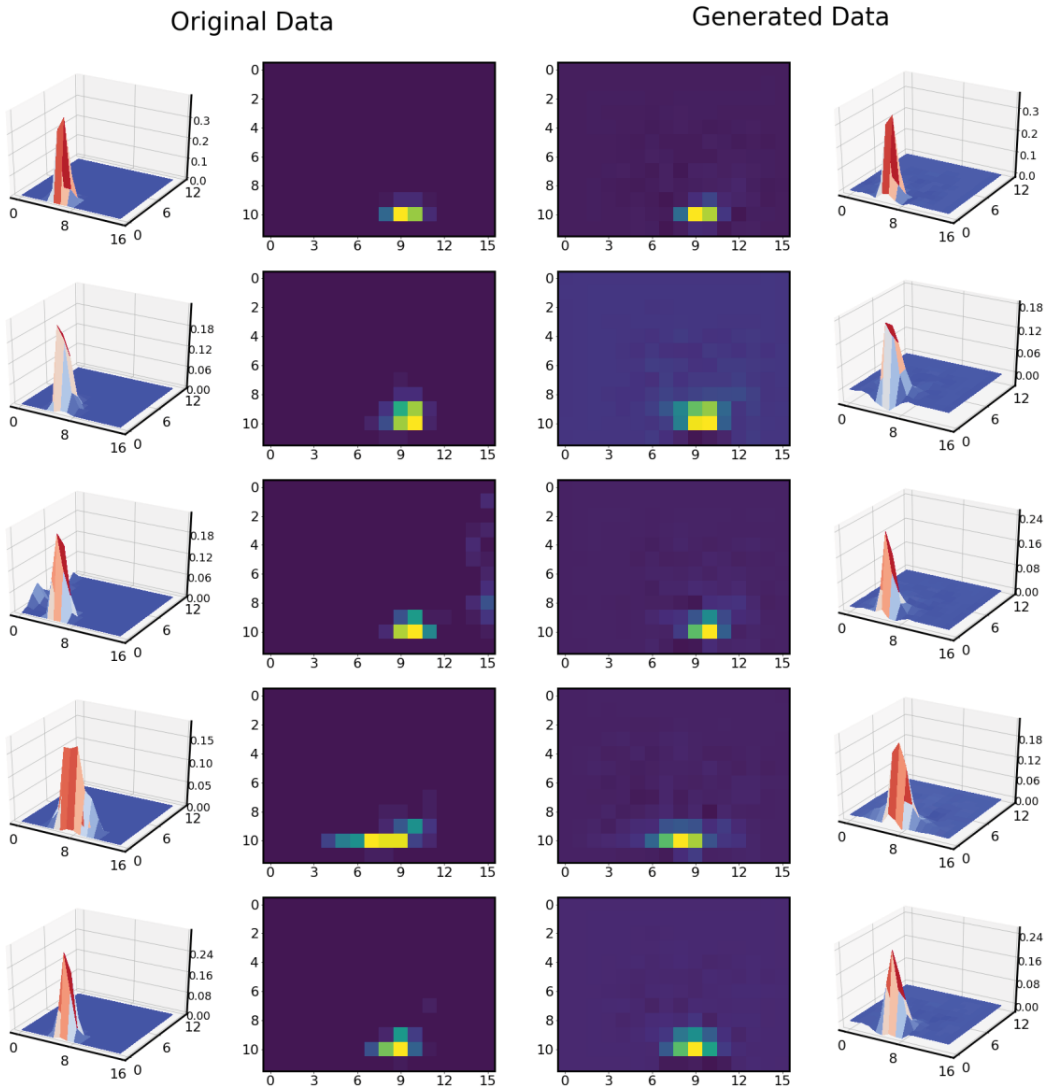

Figure 5. Comparison of original images in the test set with generated images from optimal VAE.

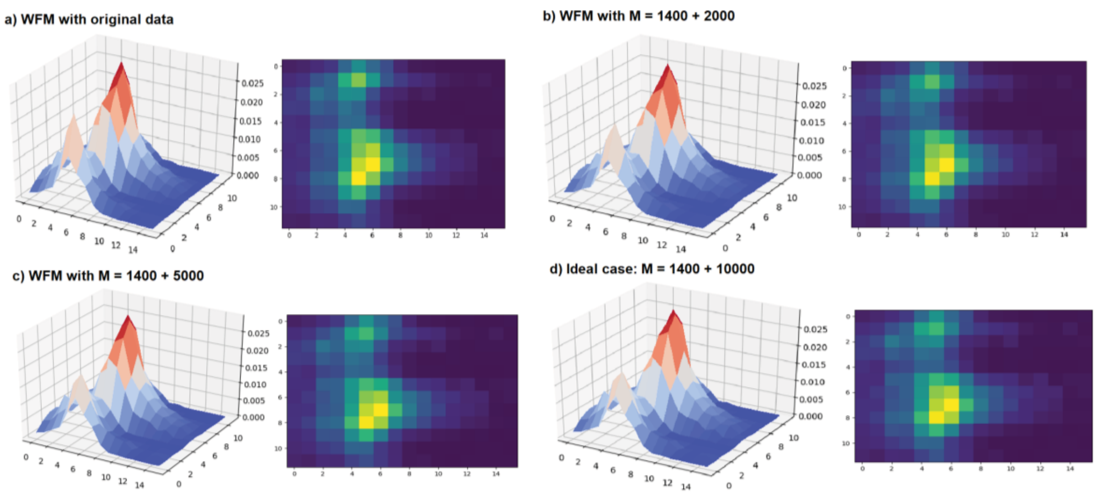

Figure 6. Wind frequency maps along with heat-map images generated for the different number of samples.

Table 2. Expected Power produced from a single turbine in 1 year using different WFMs (see Fig. 6)

S. No	Sample Size	Expected Power
1. Original data	1400	140.6 KW
2. M = 1400 + 2000	3400	148.5 KW
3. M = 1400 + 5000	6400	154.9 KW
4. Ideal case	11400	154.6 KW

In this study, the wind data needed for one year was also collected from the open source repository, ENGIE [19]. This 1-year time series of wind data, was neither used in training or testing the VAE model. Thus, this data is completely unseen in the analysis done so far and, therefore, it is considered to evaluate the power produced from a single wind turbine using the WFMs obtained in Fig. 6 and that obtained using the original data, as shown in Table 2. It can be clearly seen that the expected power produced using the WFM constructed with original data is 140.6 KW which is 14 KW less than that produced by the ideal case resulting in approximately 9% deficit. At the same time, the power obtained in the case when M = 1400 + 5000, where 5000 are the number of new WFMs generated by optimal VAE, is almost similar to that produced in the ideal case, justifying the need of generative models like optimally designed VAEs for accurate modelling of WFMS.

IV. Conclusions

In the present work, the significance of optimally designed Varitational autoencoders as deep generative networks for efficient modelling of wind characteristics is studied. The need for eliminating heuristics based hyper-parameters determination in convolutional VAEs was realized and to resolve this problem a novel multi-objective optimization based neural architecture search strategy was devised. This optimization framework, which works by balancing the trade-off between generalizability (test set accuracy) and parsimony in deep neural networks, is solved using population based evolutionary algorithm NSGA-II. Due to the lack of sufficient sequential wind characteristics data, inaccuracies crop in while building the probability mass function called Wind Frequency Map, which is used extensively in wind energy conversion systems to estimate the power produced from a wind turbine. To tackle this issue, for the first time, generative modelling has been used in this work through the optimally designed VAEs to enable accurate construction of WFMs. Further a study on the effect of number of samples on the convergence of expected power to the true value is also conducted and it was found that, WFM constructed using limited amount of data resulted in an under-estimate of expected power by 9% for a single wind turbine while that using the proposed method provided a much accurate approximation. Thus, it was found justified to implement generative modelling using optimally designed VAEs to minimize the losses incurred to wind-farmers due to such spurious estimation of WFMs.

V. References

[1] T. Burton, D. Sharpe, N. Jenkins, and E. Bossanyi, “Wind energy handbook,” Vol. 2, New York: Wiley, 2001.

[2] P. Mittal and K. Mitra, “In search of flexible and robust wind farm layouts considering wind state uncertainty,” J. Clean. Prod., vol. 248, p. 119195, 2020.

[3] H. Liu and C. Chen, “Data processing strategies in wind energy forecasting models and applications: A comprehensive review,” Appl. Energy, vol. 249, no. March, pp. 392–408, 2019.

[4] A. Almalaq and G. Edwards, “A review of deep learning methods applied on load forecasting,” Proc. - 16th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2017, vol. 2017-Decem, pp. 511–516, 2017.

[5] H. Jahangir, M. A. Golkar, F. Alhameli, A. Mazouz, A. Ahmadian, and A. Elkamel, “Short-term wind speed forecasting framework based on stacked denoising auto-encoders with rough ANN,” Sustain. Energy Technol. Assessments, vol. 38, no. June 2019, p. 100601, 2020.

[6] J. Y. Kim and S. B. Cho, “Electric energy consumption prediction by deep learning with state explainable autoencoder,” Energies, vol. 12, no. 4, 2019.

[7] A. Gensler, J. Henze, B. Sick, and N. Raabe, “Deep Learning for solar power forecasting - An approach using AutoEncoder and LSTM Neural Networks,” 2016 IEEE Int. Conf. Syst. Man, Cybern. SMC 2016 - Conf. Proc., pp. 2858–2865, 2017.

[8] S. Tasnim, A. Rahman, A. M. T. Oo, and M. Enamul Haque, “Autoencoder for wind power prediction,” Renewables: Wind. Water, Sol., vol. 4, no. 1, 2017.

[9] X. Zhao, X. Han, W. Su, and Z. Yan, “Time series prediction method based on Convolutional Autoencoder and LSTM,” Proc. - 2019 Chinese Autom. Congr. CAC 2019, pp. 5790–5793, 2019.

[10] J. Walker, C. Doersch, A. Gupta, and M. Hebert, “An uncertain future: Forecasting from static images using variational autoencoders,” In European Conference on Computer Vision, no. October, 2016, pp. 835-851,2016.

[11] G. Boquet, A. Morell, J. Serrano, and J. L. Vicario, “A variational autoencoder solution for road traffic forecasting systems: Missing data imputation, dimension reduction, model selection and anomaly detection,” Transp. Res. Part C Emerg. Technol., vol. 115, no. July 2019, p. 102622, 2020.

[12] M. Ibrahim, J. Haworth, A. Lipani, N. Aslam, T. Cheng, and N. Christie, “Variational-LSTM Autoencoder to forecast the spread of coronavirus across the globe,” 2020.

[13] F. Hutter, L. Kotthoff, & J. Vanschoren, “Automated machine learning: methods, systems, challenges,” Springer Nature, pp. 219 2019.

[14] T. Elsken, J. H. Metzen, & F. Hutter, “Neural architecture search: A survey,” 2018 arXiv preprint arXiv:1808.05377.

[15] Y. Jaafra, J. Luc Laurent, A. Deruyver, and M. Saber Naceur, “Reinforcement learning for neural architecture search: A review,” Image Vis. Comput., vol. 89, pp. 57–66, 2019.

[16] N. Nayman, A. Noy, T. Ridnik, I. Friedman, R. Jin, & L. Zelnik, “Xnas: Neural architecture search with expert advice,” In Advances in Neural Information Processing Systems, pp. 1977-1987, 2019.

[17] J. Jiang, F. Han, Q. Ling, J. Wang, T. Li, and H. Han, “Efficient network architecture search via multiobjective particle swarm optimization based on decomposition,” Neural Networks, vol. 123, pp. 305–316, 2020.

[18] K. Deb, “Multi-objective optimization using evolutionary algorithms,” John Wiley & Sons, 2001.

[19] https://opendata-renewables.engie.com/explore/dataset/39490fd2-04a2-4622-9042-ce4dd34c2a58/information

[20] Y. Bengio, I. Goodfellow, and A. Courville, Deep learning, Vol. 1, Massachusetts, USA:: MIT press, 2017.

[21] C. Doersch, “Tutorial on variational autoencoders,” arXiv preprint arXiv:1606.05908, 2016.