Heliyon 9 (2023) e15672

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Savonius wind turbine blade design and performance evaluation using ANN-based virtual clone: A new approach

Abdullah Al Noman [a,*], Zinat Tasneem [a], Sarafat Hussain Abhi [a], Faisal R. Badal [a], Md Rafsanzane [b], Md Robiul Islam [a], Firoz Alam [c]

[a] Department of Mechatronics Engineering, Rajshahi University of Engineering & Technology, Rajshahi 6204, Bangladesh
[b] Department of Mechanical Engineering, Rajshahi University of Engineering & Technology, Rajshahi 6204, Bangladesh
[c] School of Engineering (Aerospace, Mechanical and Manufacturing), RMIT University, Melbourne, Australia

ARTICLE INFO

Keywords:
Artificial intelligence
Virtual clone
SWT
Artificial neural network
Savonius blade-design

ABSTRACT

The drag based Savonius wind turbine (SWT) has shown immense potential for renewable power generation in built-up areas under complex urban wind conditions. While a series of studies have been conducted on improving SWT's efficiency, optimal performance has yet to be achieved using traditional design approaches such as experimental and/or computational fluid dynamics methods. Recently, artificial intelligence and machine learning have been widely used in design optimization. As such, an ANN-based virtual clone can be an alternative to traditional design methods for wind turbine performance determination. Therefore, the main goal of this study is to investigate whether ANN-based virtual clones are capable of determining the performance of SWTs with a shorter timeframe and minimal resources compared to traditional methods. To achieve the objective, an ANN-based virtual clone model is developed. Two sets of data (computational and experimental) are used to validate and determine the efficacy of the proposed ANN-based virtual clone model. Using experimental data, the model's fidelity is over 98%. The proposed model produces results in one-fifth the time of the existing simulation (based on the combined ANN + GA metamodel) method. The model also reveals the location of the dataset's optimized point for augmenting the turbine's performance.

1. Introduction

With a 4% contribution to global electricity generation, wind is regarded as the second-largest source of renewable energy after hydro energy [1]. Sustainable wind energy reduces dependence on fossil fuels that have a significant impact on environmental ecology and sustainable development. In 2021, the installed wind power generation capacity reached 837 GW, according to the Global Wind Report 2022 [2]. Fig. 1 depicts the exponential growth of global wind power generation capacity over time (1995–2021). On-shore, off-shore, or coastal areas seem to be the most common wind energy harnessing domains. Renewable wind energy can also be used for sustainable energy solutions in built-up or urban areas [3]. Smartly designed wind turbines can ensure the optimum use of complex and highly volatile wind energy in built-up areas.

Most commercial onshore and offshore wind turbines have production capacities in the MW range, and rural and coastal areas are suitable locations for their installation if smartly designed small-scale wind turbines are used [3,5,6]. Urban wind is erratic and

* Corresponding author.
E-mail address: nomanabdullah.mte16@gmail.com (A. Al Noman).

https://doi.org/10.1016/j.heliyon.2023.e15672
Received 19 December 2022; Received in revised form 10 April 2023; Accepted 18 April 2023
Available online 25 April 2023

Nomenclature

AI	artificial intelligence
OL	output layer
ANN	artificial neural network
RF	random forest
CFD	computational fluid dynamics
RMSE	root mean square error
C_p	coefficient of power
RVM	relevance vector machine
C_t	coefficient of torque
SSA	salp swarm algorithm
DL	deep learning
SWT	savonius wind turbine
DVM	discrete vortex method
SVM	support vector machine
DWT	darrieus wind turbine
TSR	tip speed ratio
FFT	fast fourier transform
VAWTs	vertical axis wind turbines
GA	genetic algorithm
WTs	wind turbines
GEP	genetic expression programming
A	turbine's frontal area (m^2)
GP	genetic programming
F	mechanical load on the shaft (N)
GWEC	global wind energy council
P_{Av}	available power in the wind (W)
HAWTs	horizontal axis wind turbine
P_{Tr}	output power from the turbine (W)
HL	hidden layer
T	torque (Nm)
IWO	invasive weed optimization
V	wind speed (m/s)
MAE	mean absolute error
ρ	air density (kg/m^3)
ML	machine learning
ω	angular speed (rad/s)
MLP	multi-layer perceptron
q	dynamic pressure
MSE	mean square error
R	turbine's rotational radius (m)
MSLE	mean squared logarithmic error
r	radius of the pulley on the shaft (m)
NN	neural network
N	turbine's rotational speed (rpm)
ODGV	omni-directional-guide-vane

turbulent, which makes it difficult to utilize for power generation effectively using traditional horizontal or vertical axis wind turbines. However, the vertical axis wind turbine (VAWT) is better suited to urban applications than the horizontal axis wind turbine (HAWT), because the VAWT is omnidirectional, effective at low wind speeds, and can be installed in space-constrained areas [7–9]. Among vertical-axis wind turbines, the Savonius type wind turbine is gaining popularity, despite its low efficiency due to its higher starting torque [10]. The Darrieus type wind turbine, although it has a higher power coefficient (*Cp*), is less desirable because of its greater sensitivity to approaching turbulent winds [11–13]. Darrieus turbines are more expensive because of their more complex design and despite their smaller initial torque [14,15]. Thanks to its higher reliability and capacity for self-starting, the Savonius wind turbine (SWT) is preferable over the HAWT and DWT. SWTs can operate effectively in built-up areas, including on top of buildings, in highway dividers, and near railroad tracks [3,16,17]. Wind energy systems based on the SWT are efficient for generating electricity in isolated settlements. The Savonius wind turbine has a wide range of potential and developing applications, including installation on city highways, between two tall buildings or other infrastructure, and/or alongside railway tracks [18]. Despite achieving notable

performance improvement, the full potential of the SWT has yet to be reached.

Contemporary research indicates the possible application of artificial intelligence (AI) for the advancement of the SWT blade design to enhance its performance. AI was first introduced in 1951 and is progressively used as one of the key features of modern technology and decision-making. AI is considered a central part of Industry 4.0. The concept of "artificial intelligence" is the computer simulation of human intelligence [18]. It is a growing field in many technical fields, including neurology, computer science, control systems, statistics, road vehicle aerodynamics, and battery charging framework optimization techniques [19–22]. However, a recent review paper highlighted the successful approaches of AI in diverse fields, including the possible application of AI in wind turbine optimization and power generation [18,23]. The features of artificial intelligence are numerous. The data science domain collaborates with machine learning and deep learning algorithms, two subsets of AI, to ensure smooth, inexpensive, quick, and efficient design as well as optimization and prediction approaches [18].

Existing and widely used SWT blade design or performance optimization methods include computational fluid dynamics simulation-based strategies and/or wind tunnel experiments, both of which have limitations and challenges. The computational method is costly and time-consuming in and of itself, and wind tunnel facilities are frequently unavailable and expensive. Additionally, the average wind speed is not regular in most of the places, so all the sensors installed in the experimental setup cannot operate properly on this approach. Whereas the simulation-based method, especially CFD, is expected to tackle all these issues. However, this method is considered the ideal one. The principal concern in this technique is the multi-scale turbulence and non-linearity of the Navier-Stokes fluid momentum equation. Large-Eddy Simulation (LES) and Reynolds Average Navier-Stokes (RANS) are two methods for reducing scale diversity, which are typically unrealistic and expensive due to computational resource constraints. A data-driven or machine-learning-based strategy is one way to alleviate this problem. In this study, the primary purpose is to investigate a data-driven virtual cloning method based on artificial neural networks (ANNs) that can be employed to determine the SWT's performance in terms of power coefficient under different combinations of input parameters with a shorter timeframe and fewer resources. Computational and experimental approaches take time, effort, and large resources and are costly to implement. With limited input values, the computational procedure takes a long time to yield any meaningful results. The novelty of the study is achieving fidelity with a shorter timeframe and fewer computational resources using the proposed ANN-based virtual clone model. The study is conducted based on two different cases, where Case I represents the better involvement of the proposed model compared to a prior model. The experimental data set from Case II would be used to validate the model's fidelity. The successfully developed ANN-based virtual clone model is expected to usher in significant industrial implications for wind turbine design and optimization.

2. Literature review

Numerous studies on blade profiles and augmented devices have been conducted. Even though all of these studies have shown an increasing C_p, an optimal benchmark is still lacking. Researchers have suggested that the use of AI in this regard can potentially be a suitable alternative [18]. It can be used to estimate performance and optimize a SWT's design parameters. The standard formula for calculating SWT efficiency is its power coefficient. This primary parameter depicts how well the turbine extracts kinetic energy from the wind. It is described as the ratio of the extracted wind power from the turbine rotor to the available wind power [24]. Although there are a few formulas for computing the C_p value, the fundamental equation is still the same. The C_p has always been the primary focus of turbine design, construction, and real-time implementation [25]. The torque coefficient (C_t), mechanical power and torque, tip speed ratio (TSR), drag or lift coefficients, and other basic parameters are included. The C_t and TSR have a direct impact on the power coefficient of the SWT. A higher amount of wind energy is extracted with the ideal tip speed ratio. Some of the most common and fundamental equations for C_p calculation are shown in equation 1 through 4 [26–38]. For instance, equations (1) and (4) depict the mathematical scheme of *Cp,* whereas equation (2) shows *Ct,* and equation (3) represents the TSR.

$$C_p = \frac{P_{Tr}}{P_{Av}} = \frac{T\omega}{\frac{1}{2} \times \rho A V^3} = \frac{\pi N T}{15 \rho A V^3} \tag{1}$$

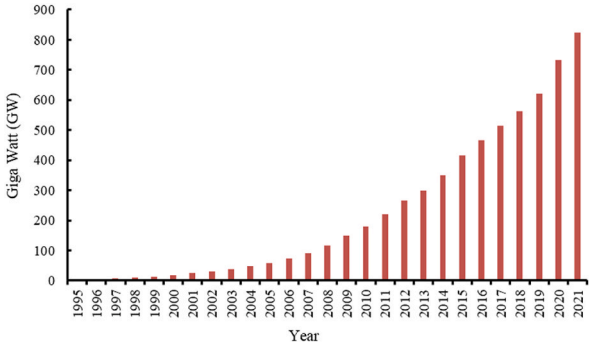

Fig. 1. Global wind power capacity growth (1995–2021) [4].

$$C_t = \frac{F \times r}{\frac{1}{2} \times \rho A V^2 R} \tag{2}$$

$$\mathrm{TSR} = \frac{\omega R}{V} \tag{3}$$

$$\mathrm{Cp} = \frac{P_{Tr}}{qAV} = C_{t\times}\mathrm{TSR} \tag{4}$$

A SWT's performance takes its particular blade profile into account. Several SWT blade profiles have been proposed. The shapes that are more concerning from 1929 include those that are semicircular, multiple quarter semicircular, splined, slotted, twisted, airfoil-shaped, Benesh, fish-ridged rotors, and helical [39,40]. All these unique profiles have attested to the substantial enhancements in turbine performance. Fig. 2 shows the advancement of the power coefficient, which corresponds to various profiles over time [[34–37, 40–51]]. However, this comparative analysis, which looks at 18 blade profiles and their output in total from 1978 to 2020, reveals that for the majority of that period, the power coefficient changes by between 0.12 and 0.33. The average *Cp* was observed at 0.217, which is not the required value, as seen in Fig. 2. Recent studies suggest that AI integration can provide a solution to resolve this issue [52–55].

The SWT augmentation techniques were used to expand another research domain with the same objective of improving the turbine's efficiency as *Cp*. Numerous studies have focused on the issue and produced unique findings. The implementation of 14 augmentation devices, including the wind shield, deflector plate, V-shaped deflector, nozzle, circular windshield, obstacle shield, curtain plates, ODGV, and others, has improved the *Cp* over the period from 1978 to 2020, as shown in Fig. 3 [35,38,40–42,46,56–63].

Although the SWT's *Cp* improves with this approach, the average value was found to be 0.277, which was somewhat higher than that of the blade profiles. Shields, guiding vanes, and curtains are a few other augmentation tools that have demonstrated an increase in *Cp* of 0.38 to 0.52. But again, this must increase the SWT's average value. According to research, AI should be incorporated to ensure that the SWT performs at its best. Additionally, recent studies have noted the involvement of AI in the dominant wind power generation sector; for example, AI has been shown to be effective at estimating wind speed [64]. As reported in the literature, a series of machine learning (ML) algorithms, including RVM, SVM, and GP, have been used to build wind turbines that can ensure better TSR. Several studies on the use of AI in wind turbine technology have previously been carried out [65–68]. Table 1 shows a summary of some of this work.

The initial sampled data can be implemented through different AI algorithms (ML or DL) along with several evolutionary and optimization algorithms to develop a digital model that can mimic traditional design approaches. The optimized model can be obtained with some small and calculative changes in hidden layers, number of neurons, and proper learning rate for a certain dataset. Fig. 4 shows the limited application of AI to SWT blade design and performance optimization, where deep learning and evolutionary algorithms are used to determine the outcomes. The virtual model can be implemented for the performance evaluation of airfoil shape, radial turbine blades, deflector geometry, and so on.

In comparison to the conventional optimization strategy, the AI-based approach is more cost- and time-effective. When using a traditional approach, complex and time-consuming design, numerical, or simulation (CFD, FFT, etc.) procedures are required. AI has

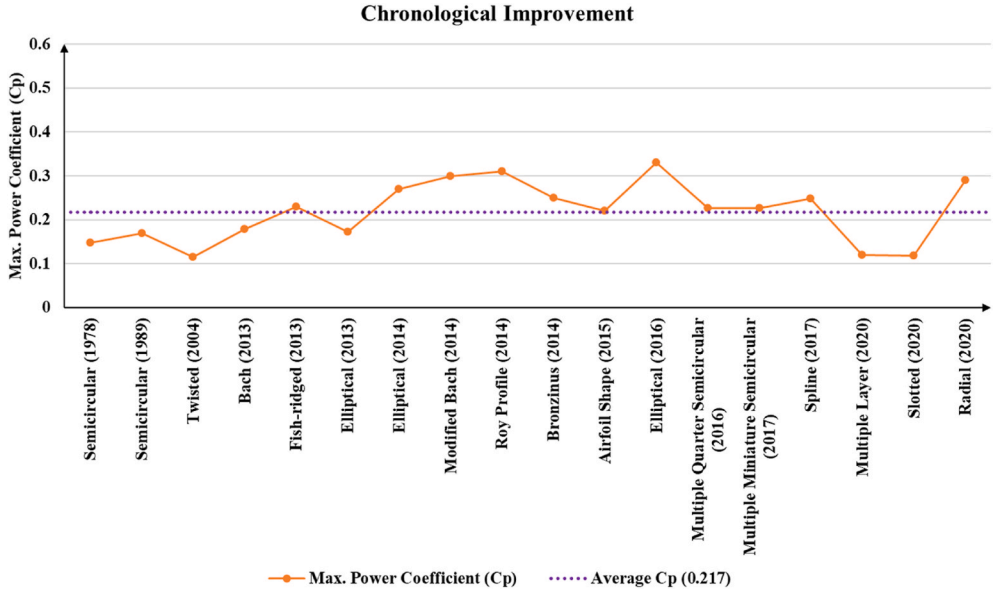

Fig. 2. The rate of change of power coefficient of SWTs with various blade profiles (adapted from Noman et al. [18]).

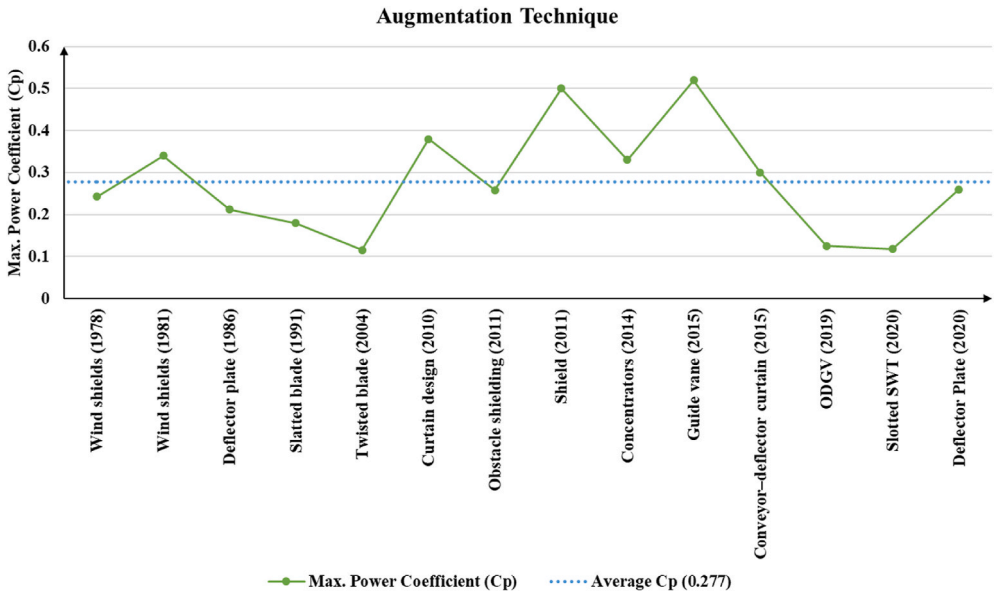

Fig. 3. The effect of different augmentation devices on the rate of change of the C_p (adapted from Noman et al. [18]).

Table 1
A glimpse of recent research on WT technology with the inclusion of AI.

Ref. & PY	Turbine Type	Dataset	Algorithm	Feature
[52] 2019	SWT	CFD Simulation	ANN, GA	Improving the efficiency by designing a deflector plate
[69] 2014	VAWT	Experimental	ANN	Turbine rotor's performance coefficient and torque coefficient estimation
[70] 2021	VAWT	Experimental	ANN	Investigating the influence of aerodynamic parameters of the turbine
[71] 2021	Not specified	Simulation	Long Short-Term Memory (LSTM)	Predicting the parameters
[72] 2022	VAWT	Simulation	Random Forest (RF)	Automatic transmission system optimization
[73] 2017	Not specified	CFD Simulation	Mean, LR, M5, RF	Wind speed prediction
[74] 2018	SWT	CFD Simulation	ANN, GA	Parameter optimization Overlap ratio, Number of stages, Blade rotation
[75] 2016	SWT	Simulation	ANN	Prediction of aerodynamic characteristics
[76] 2019	Not specified	CFD Simulation	LSTM	Wind speed forecasting and fatigue analysis
[77] 2022	Not specified	Simulation	ANN, Genetic Expression Programming (GEP)	Turbine performance estimation
[78] 2022	VAWT	Simulation	For Modeling: SVM, RF Bayesian ridge regression (BR) For optimization: Artificial Bee Colony (ABC); Genetic Algorithm (GA); Particle Swarm Optimization (PSO)	Performance optimization

made this process simpler by substituting the complex process with simple code and a set of design parameter data that requires the least amount of time, effort, infrastructure, and investment. In addition, conventionally, the entire input combination cannot be verified by simulation, whereas an AI technique verifies the appropriate output for all given data at once. Therefore, it is expected that new optimization techniques based on AI will eventually replace turbine design and optimization technology. The data validation, estimation, and optimization domains are where ANN is most frequently applied among all available ML techniques. ANN has offered a 55% enhancement ratio, considering less windy environments, by improving the proposed blade structure [79].

In order to acquire the higherC_p by ensuring blade optimization, the Salp Swarm Approach (SSA), another ML algorithm, has been

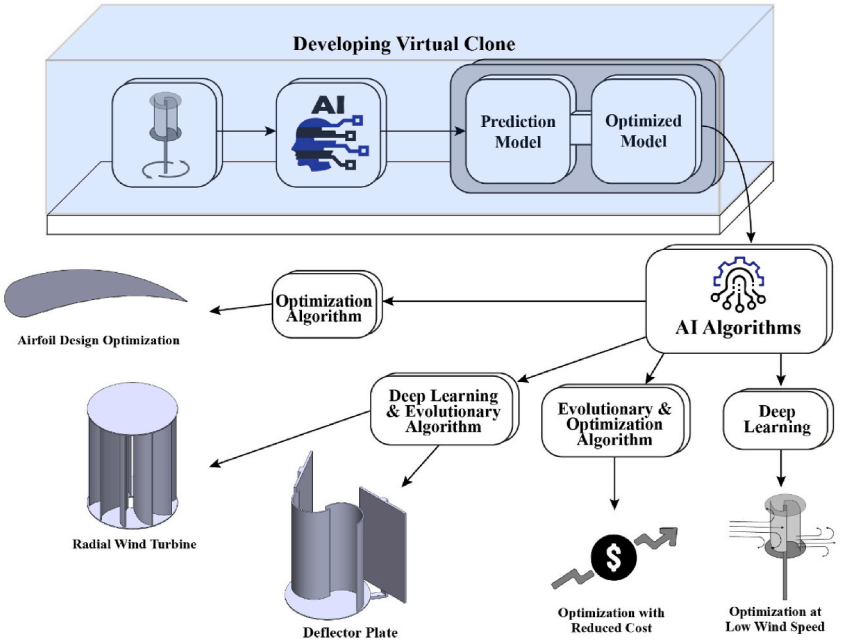

Fig. 4. AI the next generation optimizer for SWT blades.

implemented, along with the discrete vortex method (DVM) [54]. Another feature that can be used to assure blade optimization is a surrogate model based on machine learning [78]. In order to predict the power coefficient with a dataset that contains a total of 13 aerodynamic features, such as rotor diameter, overlap ratio, TSR, rotor height, etc. Rathod et al. [77] proposed a metamodel in conjunction with ANN and GEP (genetic expression programming) (for a single combination). For estimating the TSR of a VAWT featuring a deflector plate, Chen et al. [80] utilized the NN model in combination with the Taguchi method. The model performed effectively, with the error (between the CFD simulation and the projected value) being less than 4%.

Numerous ML applications have also incorporated the evolutionary algorithm (which is not an ML method) and the genetic algorithm (GA). By combining an ANN and GA metamodel with CFD simulation, Storti et al. [52] were able to optimize the deflector design and reduce costs by up to 97%. The cost function, which was created by the logical interface between the dependent and independent variables from experimental data, was analyzed and optimized using ANN and GA [74]. Again, the research [55] reported that the increase in C_p was 5.91%. The GA has demonstrated its acceptability through a number of implementations. A GA-based optimization method described in Ref. [53] led to a 33% increase in time-average Cp_{max}. Additionally, researchers [81] have presented a computational design methodology. To reduce the computational costs, they have combined the GA with IWO and non-dominated sorting-based multi-objective stochastic algorithms (NSGA-II).

Table 2 provides a brief overview of some findings from recent studies, particularly those using ML algorithms to advance SWT. As shown in the table, the application of AI outperformed all traditional methods in terms of percentage gains in power coefficient. Additionally, the strategy significantly reduces the cost of computing associated with optimization. Thus, AI can be used for turbine design and performance optimization.

From the contemporary literature review, it is evident that an artificial neural network (ANN) has immense potential for determining wind turbine performance. Although some limited studies have been undertaken (as shown in Tables 1 and 2), most of these studies are based on ideal wind conditions, and their models' validation against experimental data is not very clear. Hence, the proposed study aims to develop an ANN-based virtual clone model by validating its efficacy against experimental data with complex

Table 2
A sample of the optimization with several ML/evolutionary algorithms specially for SWT blade.

Author	Published Year	Algorithms	Observation (C_p)
Chan [53]	2018	GA	33% increase in C_p
Chern [82]	2021		5.61% increase in C_p
Shammari [79]	2020	ANN	Enhancement ratio as 55%
Mohammadi [74]	2018	ANN & GA	Cp_{max} of 0.222, with AR as 0.89 and OR as 0.159
Storti [52]	2019		30% increase in C_p
Acarer [43]	2020		3D: at TSR = 0.55 Cp_{max} is 0.29 2D: at TSR = 0.58 Cp_{max} is 0.36
Masdari [54]	2019	SSA	27% increase in C_p
Josep [81]	2020	GA, IWO, NSGA-II	Minimized computational cost

turbulence scales. Urban and built-up areas' wind conditions vary notably due to complex turbulence scales.

The proposed virtual clone model is intended to develop based on machine learning algorithms. Several digital models are to be built with ML algorithms such as decision trees, random forests, linear regression, and K-nearest neighbor (KNN) with the Case I dataset. The experimental validation of the proposed model would be based on the Case II dataset. There are some more ways to develop an optimization model with the combination of GA, IWO, and PSO (particle swarm optimization) along with ANN, which also perform well with the enriched dataset. However, the experimental dataset was small (300:6); therefore, the intended model does not concern hybrid techniques. Furthermore, the proposed ANN-based clone model has shown better performance than the previously developed model at Storti et al. [52], which combined ANN with GA. Rathod et al. [77] used the ANN model and compared it to the GEP and reported achieving better results using the ANN. The hybrid model, particularly GA-ANN, contains complex structures that necessitate more run time and computing specifications [83], and other researchers have proposed similar scenarios in which a single ANN could solve most problems.

3. Methodology

The entire workflow for this study has been discussed in several subsections. The work is split into two cases: Case I, taking into account the dataset provided by Storti et al. [52], which has a total of 300 sequences of data, each with six independent features (300:6), and Case II, taking into account the dataset of Loganathan et al. [1], with the size of 490:5. Fig. 5 depicts the entire methodology proposed to build the optimized virtual clone.

The process of developing the best ANN-based prediction model with the appropriate activation function, learning rate, number of hidden layers, optimizers, and other factors began with the collection of the dataset.

The first step is to collect and preprocess the dataset for developing the optimal sequential keras model, whose specifications are presented in Table 11. The second step (Box 1 in Fig. 5) consists of splitting both the datasets into training and test portions and the overall processes to develop, adjust (changing the activation function and optimizer), and validate the ANN-based virtual clone. Initially, the developed ANN model was trained with the training set (x_train) and validated with the corresponding x_test data to select the model's proper parameters. The optimized virtual clone is determined by considering the least model loss and best performance metrics' values after implementing a total of 8 updated activation functions and 7 optimizers. Sections 3.3 and 3.4, respectively, describe the comparative analysis of the best suit activation function and optimizer for the clone, as well as all associated results. One of the hyperparameter-tuning methods, random search, was used efficiently to determine the optimal learning rate and number of hidden layers.

As the initial performance of the proposed clone did not achieve the expected results with the prior dataset of Case II, it was then modified to enrich its volume by calculating two new features, namely "turbine speed" and "torque coefficient," from the original experimental report, and the modified dataset turns into 490:7, which results in the considerable inclination of the developed model's performance (Box 2 in Fig. 5). Lastly (Box 3 in Fig. 5), the optimized virtual clone was trained with the updated Case II dataset, and thereafter the validation dataset (x_test of Case II) of size 50:7 was utilized to evaluate the models' estimations. The predicted outputs were checked with the corresponding labelled output for calculating the performance metrics. When the clone's prediction error was observed to be the minimum, this indicated the optimized virtual clone could replicate and replace the experimental or simulation-based conventional blade design approaches. Finally, the developed model was validated with the experimental data to ensure its acceptance.

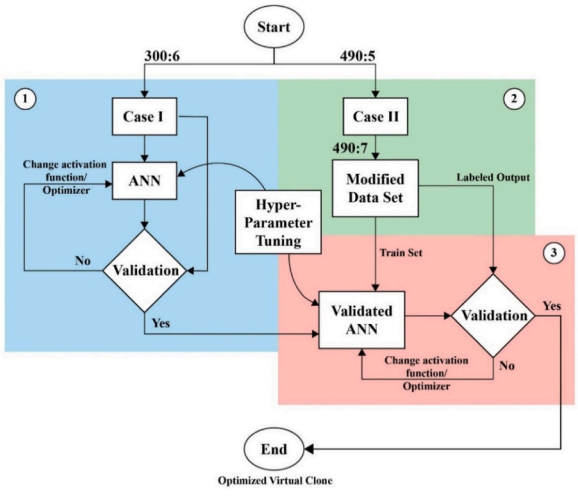

Fig. 5. Workflow of the proposed methodology.

3.1. Description of the dataset for case I

For Case I, the dataset was implemented with regard to the improvement of the SWT with a set of deflector plates. The main goal of that (presented in previous work [52]) was to determine the best deflector geometry design. The dataset was prepared initially by one of the sample variable generators, the Latin Hypercube Sampling (LHS) method. Once the sampled variables were produced, these were then tested through the CFD analysis to find the corresponding output for a certain input combination. The dataset contains six input features for a corresponding output (*Cp*), which was then utilized to train the proposed metamodel on Storti's study. A portion of the dataset with its input features and target is illustrated in Table 3.

3.2. Description of the dataset for case II

For Case II, the RMIT wind tunnel experimental dataset (Loganathan et al. [1]) was used to train and develop the best fit and optimal ANN-based virtual clone model, which can quickly replicate the experimental result. The dataset was gathered from the experimental setup of a series of prototype SWTs developed at RMIT University, Melbourne, Australia. All the prototypes' blades were semi-circular with no twisting. A custom version of Catman® AP data acquisition software was used along with a torque transducer and sensors to collect the data of three variables, namely "torque coefficient," "turbine rotational speed," and "wind speed," and other features were incorporated from the experimental setup. The number of blades varied from 8 to 48, and the wind speed was recorded from 4.5 to 8.5 m/s in the experimental dataset. Moreover, the estimated clearance also fluctuated within a range of −3 to 82 mm.

The main focus of Loganathan et al.'s [1] experiment was to determine the effect of aerodynamic parameters (number of blades, TSR, etc.), and some other influencing parameters such as the wind speed, clearance between the blades, and turbine speed on the power coefficient of the SWT. Five input features are considered in the initial dataset for the corresponding output (C_p). This dataset contains a total of 490 input combinations. Table 4 shows a section of the dataset used in the study.

3.2.1. Updating dataset of case II

Any machine learning (ML) or deep learning (DL) algorithm responds well to a significant amount of data. As the dataset for Case II had a smaller volume, certain new characteristic values from the PhD thesis by Loganathan [1] were produced from the pre-existing data by using the governing equation to increase the volume of the dataset. Two of the additional elements, "turbine speed" and "torque coefficient," were manually added to enhance the model's performance. The procedure was carried out in such a way that it left the other parameters unaffected. The correlation matrix of the dataset in Fig. 6 has demonstrated the independence of all the features. With one exception (turbine speed, TSR), all of the characteristics appear to be independent because their values in the correlation matrix are all considerably less than 1. However, as the newly added features have enhanced the model's performance, this case can be neglected.

The dataset has been slightly modified from the original dataset. The adjusted dataset is shown in Table 5. The modified dataset's csv (comma separated value) file was first uploaded to colab. The required libraries were then added, and some file installations were also performed. The keras sequential technique was used to create the ANN-based prediction model. The random search technique was used to tune the hyperparameters.

3.3. Choice of activation function and optimizer for proposed clone (case I and case II datasets)

The primary goal of establishing the ML/DL model was to replace the traditional methodologies for estimating the performance of turbine blade designs. Only the balanced or best-fit model can accurately reproduce the simulation or experimental process. The optimizer, activation function, learning rate, and other model specifications make up a balanced NN model. In the process of creating

Table 3
Portion of the Case I dataset with its input features and target.

r_1 (cm)	Θ_1 (°)	Δr_2 (cm)	$\Delta\Theta_2$ (°)	Δr_3 (cm)	$\Delta\Theta_3$ (°)	C_p
137	10	19	4	8	4	0.161
139	5	15	2	13	4	0.143
139	7	7	3	17	4	0.147
148	4	24	1	7	5	0.159
147	9	9	3	29	2	0.168
138	12	28	5	20	4	0.192
133	10	26	5	13	4	0.182
133	6	18	2	16	4	0.167
150	6	30	4	26	4	0.184
135	8	17	2	14	3	0.165
145	14	28	2	8	5	0.176
136	5	8	4	19	3	0.145
152	1	9	2	25	1	0.175

There was a total of 300 input combinations for the entire arrangement. The same dataset was implemented here in this work for developing the virtual clone initially: r and Θ ($\mathbf{r_1}$, $\mathbf{\Theta_1}$, $\mathbf{\Delta r_2}$, $\mathbf{\Delta\Theta_2}$, $\mathbf{\Delta r_3}$, $\mathbf{\Delta\Theta_3}$) represent the linear distance from the centre of the rotor and the angle difference of the deflector plate, respectively.

Table 4
A portion of the initial dataset used in Case II

Blade number	Wind speed (m/s)	TSR	Predicted clearance (mm)	Clearance error (%)	C_p
8	8.5	0.0013	82.0	1.23	0.221
8	8.5	0.0051	82.0	1.23	0.203
8	8.5	0.0078	82.0	1.23	0.185
8	8.5	0.0095	82.0	1.23	0.166
8	8.5	0.0105	82.0	1.23	0.148

Fig. 6. Correlation matrix of the Case II dataset.

Table 5
Some modified dataset for Case II provided by Loganathan [1].

Blade number	Wind speed (m/s)	TSR	Turbine speed (rad/s)	Torque coefficient	Predicted clearance (mm)	Clearance error (%)	C_p
8	8.5	0.0013	0.719	166.83	82.0	1.23	0.221
8	8.5	0.0051	2.736	40.20	82.0	1.23	0.203
8	8.5	0.0078	4.196	23.83	82.0	1.23	0.185
8	8.5	0.0095	5.153	17.47	82.0	1.23	0.166
8	8.5	0.0105	5.658	14.14	82.0	1.23	0.148

the virtual clone, a detailed analysis of the best optimizer was performed by the trial-and-error method (sequential model). The list of optimizers and their corresponding results for the dataset are shown in Table 6. It was quite difficult to choose the best optimizer with the chosen HL, and learning rate because the dataset (used in Case II) is smaller in size. As of yet, "Adam" was the model's best match for the dataset used in this study.

The summary of the analysis is displayed in Table 6, following comparison with other optimizers. The loss function MSE was 6.6585×10^{-3} and various other metrics were noted, along with the optimizer "SGD", which was used to begin the comparison. The MSE for the remaining optimizers, including "RMSprop", "Adadelta", "Adagrad", "Adamax", and "Nadam", was 3.3021×10^{-3}, 2.2734×10^{-2}, 4.4467×10^{-2}, 1.816×10^{-3}, and 3.0967×10^{-3}, respectively. While **"Adam"** revealed the lowest MSE to be $\mathbf{1.0807 \times 10^{-3}}$ and the lowest value for the other performance metrics (r2 score, MAE, and RMSE), which were used as shown in Table 6.

Additionally, the performance of a NN model depends on the appropriate selection of the ideal activation function for the specific dataset. A detailed comparative analysis has been undertaken following the same method (as optimizer selection) in order to identify the best activation function for constructing the virtual clone. For the datasets of Case I and II, respectively, Tables 9 and 10 show the overall and detail results represented at Section 4.1.1.

Table 6
Investigation of the virtual clone's performance using several optimizers in comparison. (Containing the learning rate as 0.001, the optimizer as Adam, the maximum number of epochs as 500, the activation function for the output layer as linear, and the number of HL as 7).

Optimizer	Accuracy Train/Val	R2_Score	MAE	MSE	RMSE
SGD	97.14/96.82	74.0	0.063	6.6585×10^{-3}	0.0816
RMSprop	96.88/95.26	87.1	0.042	3.3021×10^{-3}	0.0575
Adadelta	90.36/95.26	11.1	0.109	2.2734×10^{-2}	0.1508
Adagrad	96.09/95.26	82.6	0.050	4.4467×10^{-2}	0.0667
Adam	**97.40/96.88**	**95.8**	**0.016**	$\mathbf{1.0807 \times 10^{-3}}$	**0.0329**
Adamax	96.88/95.26	92.9	0.020	1.816×10^{-3}	0.0426
Nadam	96.00/96.00	87.9	0.039	3.0967×10^{-3}	0.0556

Table 7
Specification of the optimal ANN topology for case I and case II datasets.

Parameters	Specification (case I)	Specification (case II)
Input	6	7
Output	1	1
HL of neural network	3	3
Drop out	0.2	0.2
No. of epochs	1500	500
Mean Absolute Error	0.0037	0.016
Mean Squared Error	1.353×10^{-5}	1.081×10^{-3}
RMSE	2.98×10^{-3}	0.033
R2_Score	93.9	95.8
Optimizer	Adam	Adam
Learning rate	0.001	0.001
Activation function for HL	elu	softplus
Activation function for output layer	linear	linear

Table 8
Some formula is used for the performance metric calculation [87,88].

Evaluating metrics	Governing formula	Where,
R2_score	$1 - \frac{ss_{residual}}{ss_{total}}$	Residual = (observed value – fitted value) $SS_{residual}$ = sum of the squares of residuals SS_{total} = total sum of the squares y_{hat} = predicted value by the model y_{tr} = target output
MAE	$\frac{1}{n}\sum_{i=1}^{n} y_{tr} - y_{hat}$	
MSE	$\frac{1}{n}\sum_{i=1}^{n} (y_{tr} - y_{hat})^2$	
RMSE	$\sqrt{\frac{1}{n}\sum_{i=1}^{n} (y_{tr} - y_{hat})^2}$	

Table 9
Comparative performance analysis of the virtual clone with several activation functions for Case I

Activation Function for HL	Accuracy Train/Val	R2_Score	MAE	MSE	RMSE
relu	1.00/1.00	92.40	0.0031	1.6823×10^{-5}	0.0041
sigmoid	1.00/1.00	−0.011	0.0122	2.2471×10^{-4}	0.0149
softplus	1.00/1.00	87.70	0.0040	2.7403×10^{-5}	0.0052
softmax	1.00/1.00	−0.008	0.0122	2.2348×10^{-4}	0.0149
softsign	1.00/1.00	90.90	0.0034	2.0229×10^{-5}	0.0045
selu	1.00/1.00	72.10	0.0069	6.1980×10^{-5}	0.0079
elu	**1.00/1.00**	**93.90**	**0.0028**	$\mathbf{1.3529 \times 10^{-5}}$	**0.0037**
tanh	1.00/1.00	85.70	0.0047	3.1863×10^{-5}	0.0056

Table 10
Comparative analysis of the performance of the virtual clone with several activation functions for the Case II dataset (Containing the learning rate as 0.001, the optimizer as Adam, the maximum number of epochs as 500, the activation function for the output layer as linear, and the number of HL as 7).

Activation Function for HL	Accuracy Train/Val	R2_Score	MAE	MSE	RMSE
relu	97.14/95.26	93.5	0.016	1.6524×10^{-3}	0.0407
sigmoid	97.14/96.82	95.6	0.012	1.1302×10^{-3}	0.0336
softplus	**97.40/96.88**	**95.8**	**0.016**	$\mathbf{1.0807 \times 10^{-3}}$	**0.0329**
softmax	94.27/95.26	−0.037	0.140	2.6509×10^{-3}	0.1628
softsign	97.66/95.31	94.1	0.021	1.5106×10^{-3}	0.0385
selu	96.88/95.26	93.3	0.024	1.7026×10^{-3}	0.0413
elu	97.40/96.88	94.4	0.019	1.4438×10^{-3}	0.0379
tanh	97.92/95.26	93.9	0.022	1.5499×10^{-3}	0.0394

3.4. ANN architecture

Along with some ML techniques (hyperparameter tuning, linear search, etc.), a virtual clone based on artificial neural networks (ANNs) has been developed to mimic any experimental model and CFD as a balanced model (with less mean absolute error) and to find the dataset's optimal point, ensuring the highest *Cp*. In this study, a supervised machine learning model has been created, etc.), a

Table 11
Comparison of the proposed model with Storti's model in Case I.

Parameters	Storti's model	Proposed model
Input	6	6
Output	1	1
Training data size	300	300
Validation data size	50	50
HL	1 to 3	3
Activation Function for HL	Log-sigmoid	elu
Activation Function for OL	Log-sigmoid	linear
No. of epochs	1500	1500
MSE	$\mathbf{2.154 \times 10^{-4}}$	$\mathbf{1.353 \times 10^{-5}}$
RMSE	———	**0.0037**
MAE	$\mathbf{7.7 \times 10^{-3}}$	$\mathbf{2.98 \times 10^{-3}}$
R2_Score	**97**	**93.9**
MSLE	———	$\mathbf{1.4 \times 10^{-5}}$

virtual clone based on artificial neural networks (ANNs) has been developed to mimic any experimental model and CFD as a balanced model (with less mean absolute error) and to find the dataset's optimal point, ensuring the highest *Cp*. In this study, a supervised machine learning model has been created. When an ML model uses labelled data, all possible input combinations for the training data's outputs are known. The ANN model can estimate the best potential output from any unknown input combinations with the right learning of the dataset. In 1943, Warren et al. [84] proposed the perceptron model for supervised techniques and the use of the neural network in non-linear modelling [77].

Fig. 7 shows the very basic model structure of an ANN. According to this figure, the artificial neurons in this case have synapses, an activation function, a summation point, and a bias. The intermediate links between the input and summation nodes are referred to as synapses, also referred to as weights. The input variables and synaptic weight are combined linearly via the summing node (Σ) (multiply them and add them together). The summing node typically outputs a greater number because of its internal calculation; in this case, an activation function φ(.) should be employed to determine the output within a specific range. The bias is also used to interrupt the summing value at node Σ (by increasing or decreasing).

Both single-layer and multilayer perceptron can be used in the ANN (MLP). The MLP is very suitable for many issue analyses, including multivariable regression, fitting, time series analysis, etc., according to the most recent research [85]. It has been established that MLP is a general function approximator with specific characteristics [86]. The non-linear multivariable input-output problem analysis that may be carried out using the MLP paired with the back propagation technique is best suited for the suggested thesis domain [85]. Between the input and output layers, the MLP architecture has one or more hidden layers. The number of hidden layers, learning rate, node count, and many other variables pertaining to the problem type can all be determined using a variety of techniques. The hyperparameter tuning methodology has been applied in this work, utilizing the random search method. Although one hidden layer is seen as the ideal amount for multivariable regression problems, using too many hidden layers can result in increased local minima or unstable gradients [77].

Therefore, a fully connected MLP with two hidden layers has been used in the current research area. An MLP neural network that is fully connected is shown in Fig. 8. One input layer, one output layer, and two hidden layers are shown in this diagram. The input layer links the hidden layers to the provided input variables in a similar manner to how the synaptic connection between the hidden layer and output layer is constructed.

In the present work, the Google CoLab, an online machine learning, deep learning, and data analysis platform with Python code execution, was used to incorporate the proposed approach. A total of 56 ANN-based virtual clones were initially developed with the mentioned activation functions and optimizer, and finally the best-fit model was determined by selecting the optimal model specification through trial and error (for choosing the activation function and optimizer) and random search (for obtaining proper HL and learning rate).

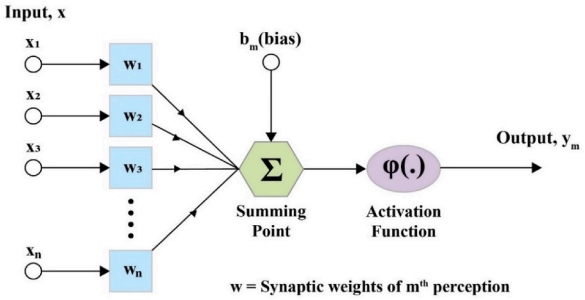

Fig. 7. Block diagram of a basic ANN architecture.

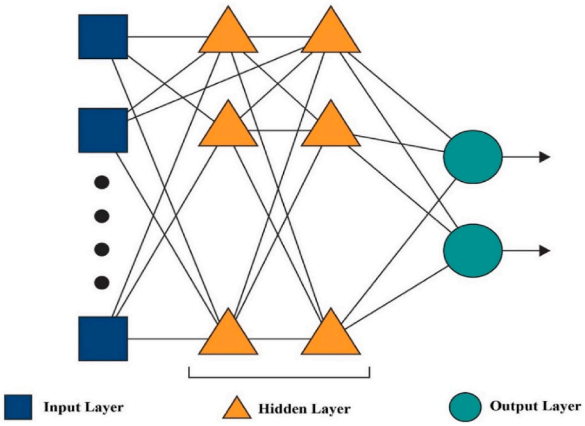

Fig. 8. Multilayer perceptron with ANN model.

3.4.1. Hyper-parameter tuning (random search)

An AI-based model's performance depends on its optimal specification. The appropriate selection criteria reduce the NN complexity and ensure smooth execution with minimal execution time. The best-bit model needs to acquire the appropriate number of neurons, hidden layers, and learning rate for the specific dataset. Manual selection of these mentioned parameters' values cannot ensure an exact match; moreover, these are time-consuming, whereas the hyperparameter tuning approach has been used to determine the optimal learning rate, number of HL, and number of nodes in the hidden layers. Compared to the conventional trial-and-error method, this method is quicker and more efficient in confirming the model's best performance. This technique is reliable for selecting the convenient NN model's parameter. In this study, the best fit HL and learning rate were determined as 3 and 0.001, respectively, by the automated approach.

To declare the best optimizers for the model, trial-and-error was used to study the activation functions. A thorough comparative analysis of the set of activation functions and optimizers has been undertaken. Table 7 shows the optimized specification of the developed ANN-based virtual clone for both the case I and case II datasets, with which the developed models produce nearly the same output as the experimental values. Additionally, the dropout of 0.2 was used as the regularization of the proposed clone to ensure a balanced model free from underfitting or overfitting issues. Following that, the best-fitting and most balanced virtual clone was used to test its adaptation to the feeding data on the mentioned dataset. The models' best performance is conspicuous by the value of the individual performance metrics addressed in this analysis. The least prediction error (MSE, RMSE, and MAE) was confirmed later by the improved virtual clone. For Case II data, the created model validated the correctness at 98.44 (%), with a coefficient (R_Squared) value of 95.8, a MSE of 1.081 10^{-3}, a RMSE of 0.033, and a MAE of 0.016, as described in Table 12. The least error value is the best outcome from this model, which was much better and reasonably close to the labelled output. Moreover, the comparative analysis with Storti's model is listed in Table 11.

4. Results and discussion

For both the Case I and Case II datasets, a neural network (NN) model has been created with various properties. The finest selective (compared) parameters, such as the activation function, learning rate, and optimizer, were used to construct the optimal model, and their best-matched value was assigned using extensive comparative analyses. The optimized virtual clone has confirmed the most efficient values for all the mentioned performance evaluation metrics. Among several evaluating metrics, the RMSE, MSE, MAE, and r2_score have been examined in this current work, which also performed the specific governing formula behind each of their individual

Table 12
Performance of the proposed model for Case II dataset.

Parameters	Specification
Training/Validation percentage	97.40/96.88
No. of epochs	500
Mean Absolute Error	0.016
Mean Squared Error	1.081×10^{-3}
RMSE	0.033
MSLE	0.001
R2_Score	95.8
Optimizer	Adam
Learning rate	0.001
Activation function for HL	softplus
Activation function for output layer	linear

calculations. Table 8 illustrates some of the most common and fundamental governing equations behind their calculations.

4.1. Performance visualization for optimal activation function

The performance of the proposed virtual clone model is further visualized with each of the individual activation functions, and the best-fit model is then determined considering the model accuracy and loss curve along with the performance metrics. Moreover, the corresponding accuracy and loss curves have been plotted for determining the number of epochs needed to find the best-fitting or most balanced model. With the Case I dataset, the activation function "**elu**" and the optimizer "**Adam**" demonstrated the best performance, as demonstrated in Table 9 in detail, whereas the Case II dataset revealed the best performance with the activation function "**softplus**" and the optimizer "**Adam**". They represented it in Table 10.

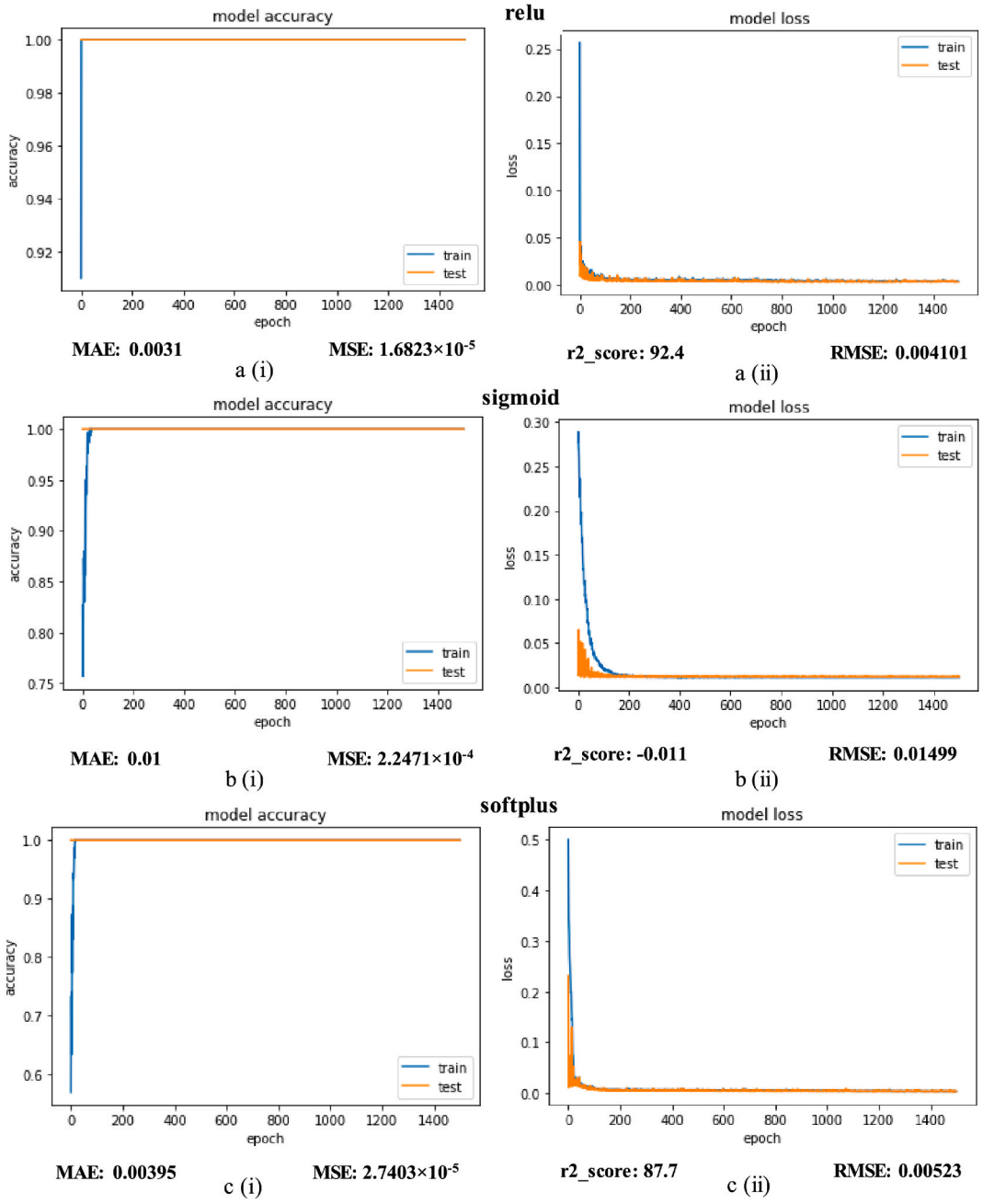

Fig. 9.1. Model accuracy and loss plot for Case I with different activation functions (a) "relu" (b) "sigmoid" (c) "softplus".

4.1.1. Optimal activation function for case I

Following the performance metric values in Tables 9 and it is evident that among all cited activation functions, "**elu**" confirmed its best suitability for the corresponding dataset (Case I). The investigation was done with the prior specification, considering the mentioned HL, learning rate, and optimizer, and the optimal activation function was then determined. The difference among all the functions addressed gets conspicuous in the graphical representation. For accuracy, all graphs are shown from Fig. 9.1(a) i to 9.1(h) i, whereas for model loss as the number of epochs changes, this is evident from Fig. 9.1(a) ii to 9.1(h) ii.

The graphs show the connection between the virtual clone's test and training performances. Although all of the activation functions used in this study have roughly the same shapes in terms of model correctness, the associated activation functions have seen very little shape change due to loss.

Even though there was not a particularly noticeable shift in the loss function, in fact the activation function "**elu**" in Fig. 9.1(g) ii

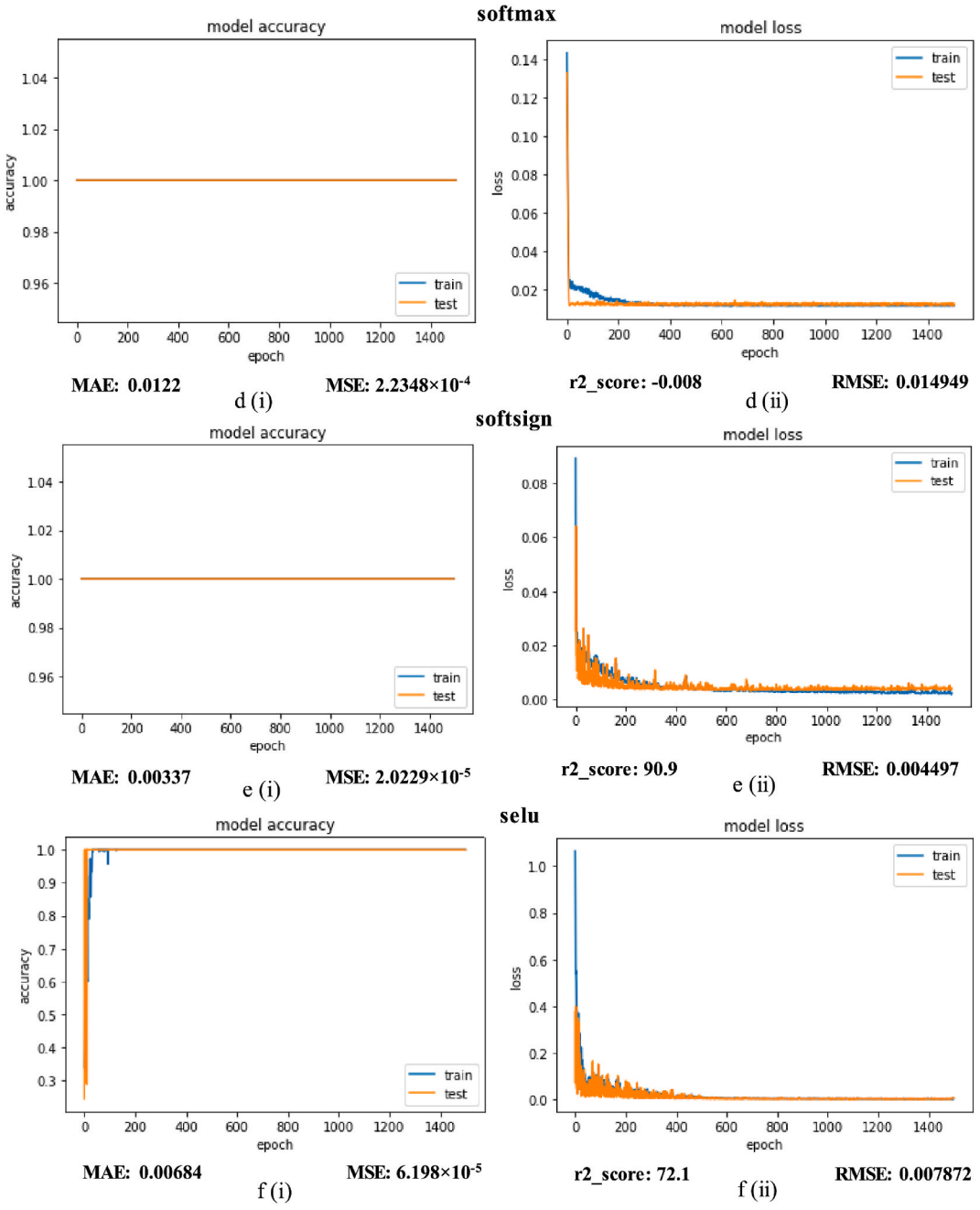

Fig. 9.1. Model accuracy and loss plot for Case I with different activation functions (d) "softmax" (e) "softsign" (f) "selu".

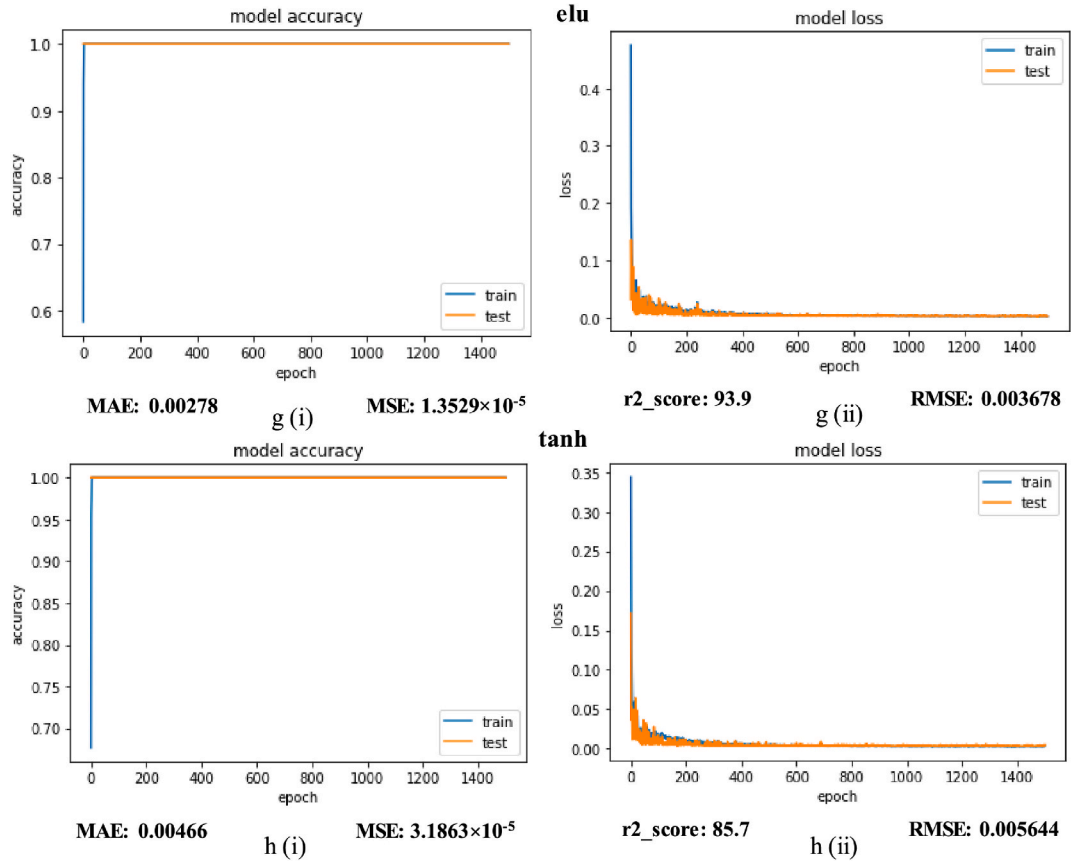

Fig. 9.1. Model accuracy and loss plot for Case I with different activation functions (g) "elu" (h) "tanh".

has produced the best r2 score with a smooth change in the model loss. Consequently, with the particular dataset, the function "**elu**" can be declared to be the most appropriate for all calculated performance metric values. Additionally, "**elu**" verified the virtual clone's effective implementation, as shown in Fig. 10. The relationship between the predicted and actual values depicted on the graph is quite similar. Again, the estimated curve overlaps the target curve with most of the data points, demonstrating the virtual clone's excellent adaptability to the dataset.

The activation function "**elu**" is shown to have the least model loss, which was noted in the loss function MSE as $\mathbf{1.3529 \times 10^{-5}}$ from the constructed model. All other functions listed, such as "sigmoid," "softplus," "softmax," "softsign," "selu," and "tanh," have been observed with their respective MSEs of 2.2471×10^{-4}, 2.7403×10^{-5}, 2.2348×10^{-4}, 2.0229×10^{-5}, 6.198×10^{-5}, and 3.1863×10^{-5}. The activation function "relu" has ensured its MSE of 1.6823×10^{-5}. For all executed activation functions, Table 9 includes additional performance metric data.

4.1.2. Optimal activation function for case II

For Case II, the sequential keras model's optimal activation function and optimizer were chosen using the same analysis as was done for Case I. Eight activation functions in all have been added to this model; the list, together with their accuracy and corresponding loss (performance metrics), is shown in Table 10. After a thorough comparative analysis (trial-and-error) represented in the table, it can be stated that the activation function "**softplus**" shows the best performance with the corresponding dataset (Case II) among the other listed functions.

To visualize the models' execution with that of the Case II dataset, a couple of sudden fluctuations have been observed, which are shown in Fig. 9.2(a) i to 9.2(h) i for the accuracy and Fig. 9.2(a) ii to 9.2(h) iifig10a for the model loss with the change of number of epochs.

The model accuracy was found with the activation "relu" at 98.7% for the training set and 98.44% for the test set, whereas the accuracy with "sigmoid" was (97.4/96.88) as a train/test percentage. This represents an abrupt change on the accuracy graph with the respective activation functions. The "softmax" has verified its accuracy as 94.97/95.25, demonstrating the virtual clone's consistent and well-balanced performance. The accuracy of the functions "softsign," "selu," "elu," and "tanh" has been verified as 97.66/95.31, 95.83/95.31, 97.40/96.88, and 97.66/95.26, respectively. The maximum model accuracy, as determined by the activation function "softplus," was **98.44/98.44**, highlighting the fully balanced model in comparison. This model also confirmed the lowest MSE, which was $\mathbf{1.2602 \times 10^{-3}}$. Table 10 includes the values for MSE and other performance metrics for each related function.

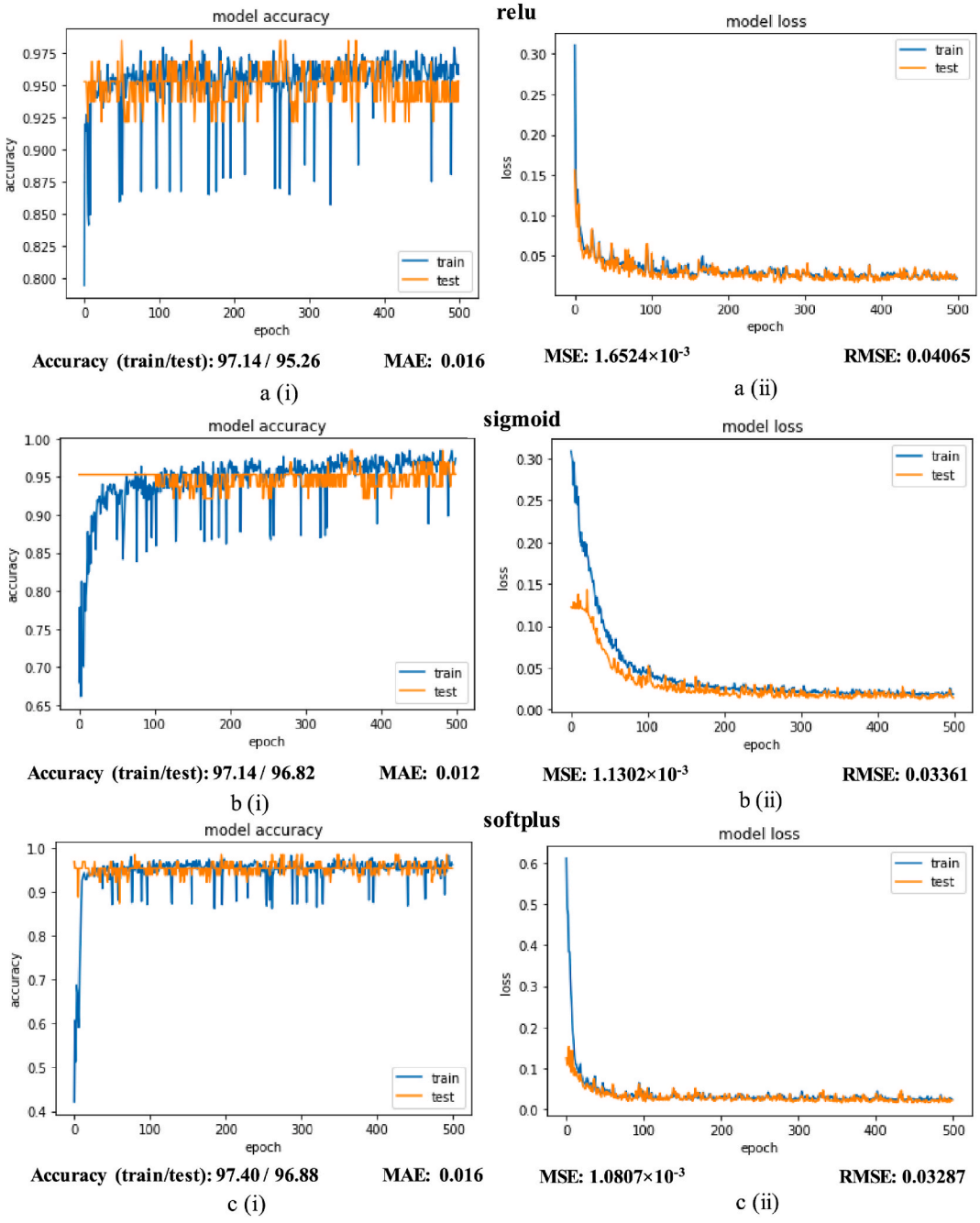

Fig. 9.2. Model accuracy and loss plot for Case II with different activation functions (a) "relu" (b) "sigmoid" (c) "softplus".

The best-fit and most balanced model for case II has been confirmed by Fig. 9.2(c) with the function "**softplus**". In this instance, as opposed to others, the model's accuracy with the train and test response was observed to be smooth. Additionally, the activation function has validated the best value of the various performance metrics. Finally, as shown in Table 10, "**softplus**" for both hidden layers and "**linear**" for the output layer provided the best fit model for the specific dataset of all the activation functions used to develop the keras sequential model. Fig. 12, with the function "softplus," illustrates the relation between the actual and predicted values of Case II. The majority of the data points on the predicted curve overlap the target curve, demonstrating the virtual clone's perfect match with the dataset.

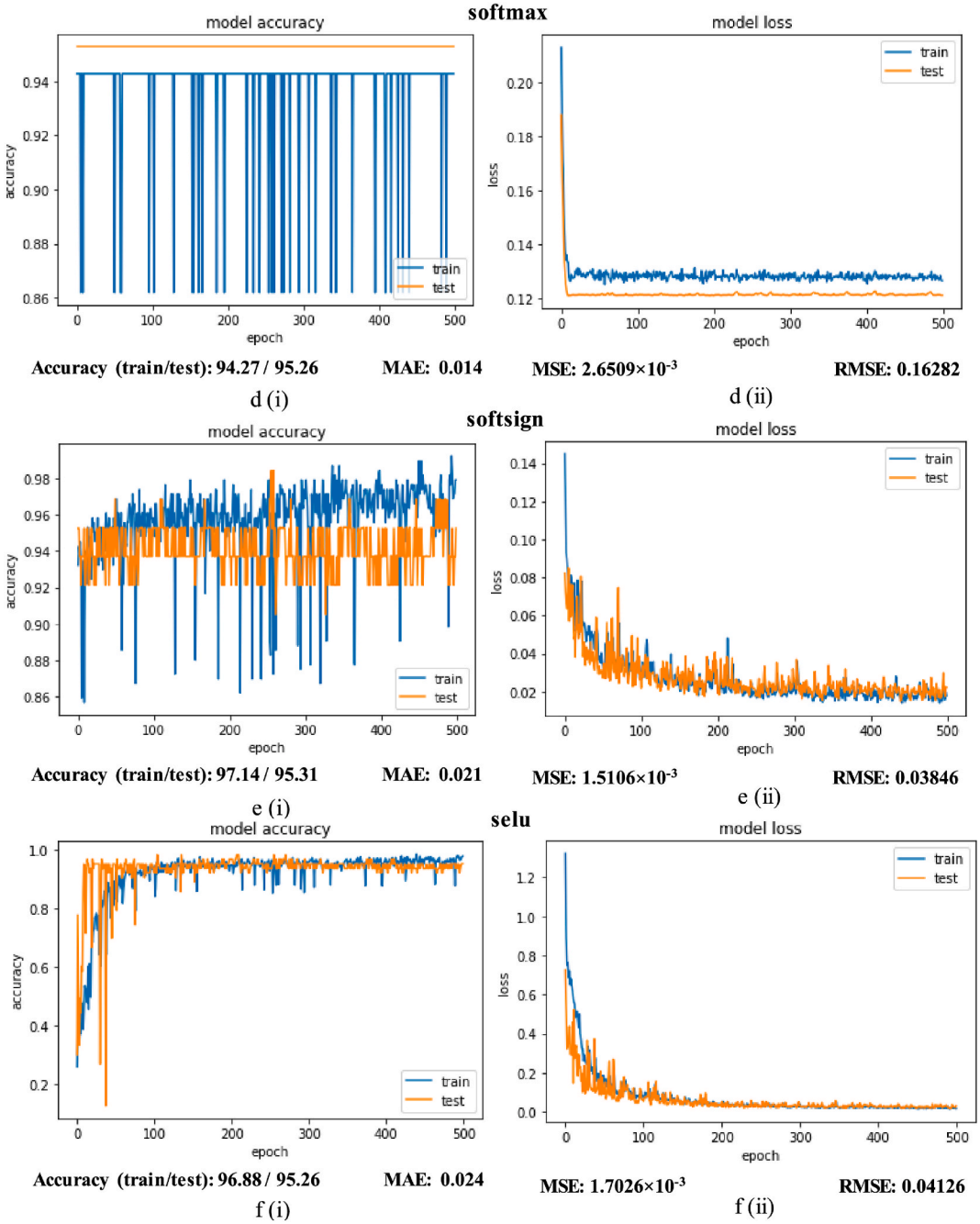

Fig. 9.2. Model accuracy and loss plot for Case II with different activation functions (d) "softmax" (e) "softsign" (f) "selu".

4.2. Comparative analysis of the proposed model with Storti's model

With the Case I dataset as its initial implementation, the suggested virtual clone performed significantly better than the earlier work described in Storti et al. [52]. All performance metrics were cited in this study with superior results. Table 11 provides an illustration of the comparative analysis.

The proposed model is best suited for the given dataset after making just a few minor changes to the model specifications. Apart from the r2 score, where the change is not very great, performance metrics as shown in Table 11 appeared to be substantially better with the proposed model. Additionally, this clone model is ideal because the MSE is 1.353×10^{-5} and the MAE is 2.98×10^{-3} with a modified optimizer and activation function. As a result, when all evaluation metrics are taken into account, the virtual clone has been found to be the best alternative to the model described in the earlier work [52].

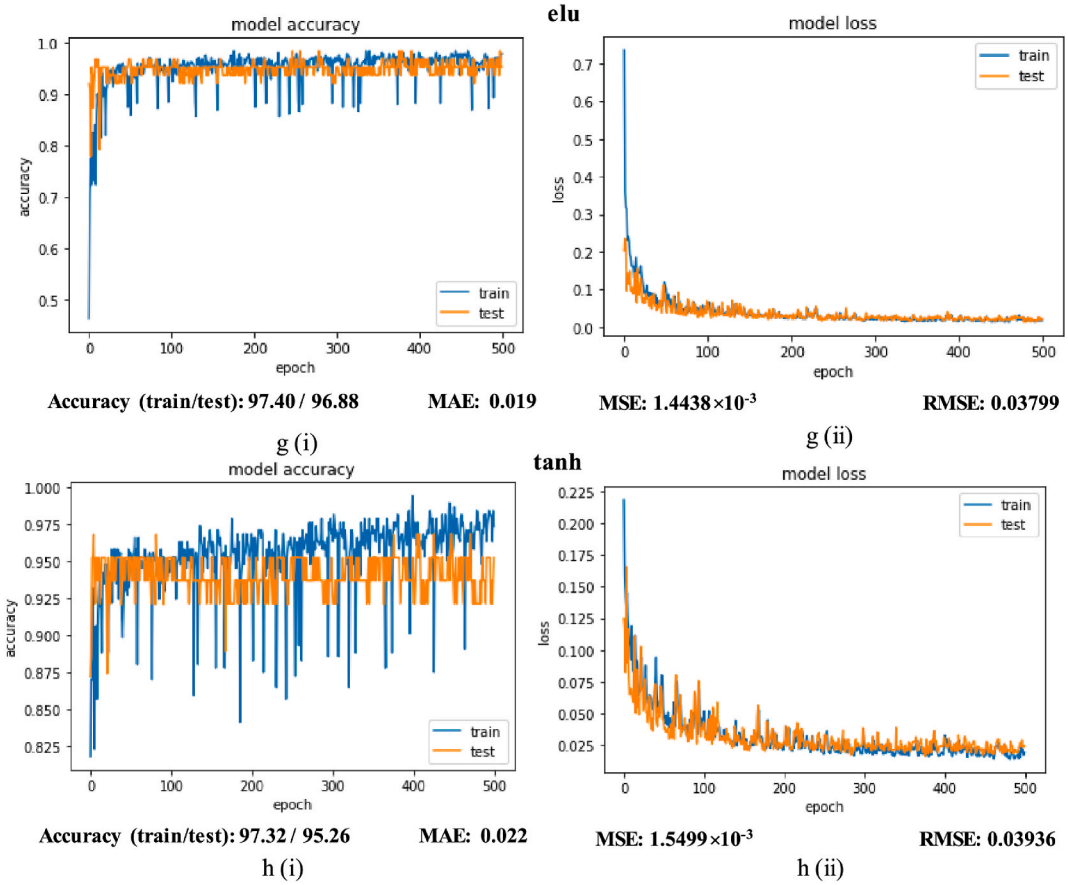

Fig. 9.2. Model accuracy and loss plot for Case II with different activation functions (g) "elu" (h) "tanh".

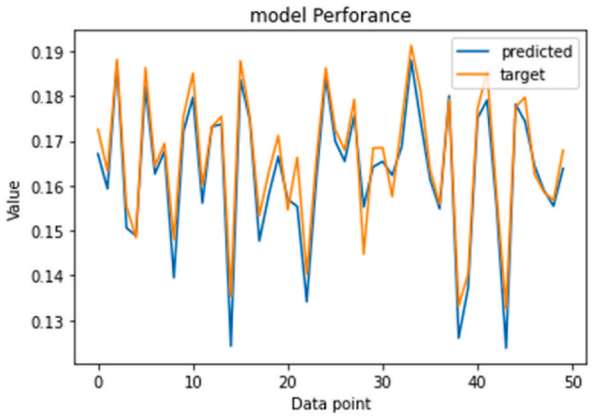

Fig. 10. Virtual clone's performance under optimal specification for Case I dataset.

4.3. Performance of the virtual clone with the case II dataset

The overall specification as well as the outcomes (evaluation metrics) of the developed virtual clone are shown in Table 12. As the experimental dataset has been used to construct a clone model for the very first time, the performance of the NN models is the benchmark. The suggested model can assist in developing a balanced model and prevent problems with overfitting and underfitting. After the thorough analysis, it can be declared that the generated model reported in this work was fully balanced with the appropriate activation function, learning rate, optimizer, and other variables.

Among 56 ANN-based virtual clones, the best suit model is selected as the optimized model, considering the bright match of the

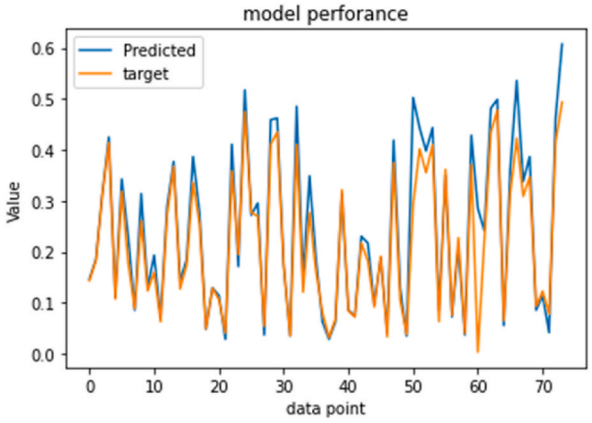

Fig. 11. Virtual clone's performance with optimal specification for Case II dataset.

activation function, optimizer, and learning rate described in Table 7. The final model indicates the best engagement with the experimental sample dataset for both cases compared to all the previously developed models. For Case I, the developed clone leads all the mentioned performance metrics of the prior Storti model except the r2_score, which is still a close value. All the other metrics recorded are much better than the previous model compared in Table 11. Thus, the proposed model in this work outweighs the performance of the prior metamodel. Again, with Case II, the substantially reduced loss functions and higher gain values confirm the optimized virtual clone described in Table 12.

4.4. General discussion

The "linear search" machine learning technique was used to provide an additional feature to the virtual clone. From the whole dataset, the generated clone can identify the input combinations from which the SWT derives the best value for C_P. For the enriched dataset, where the manual finding approach may be difficult or time-consuming, this process can provide a good solution. For the Case II dataset, the power coefficient of the SWT was determined from the x_test dataset and then compared with the labelled values. From the entire test data, the best three input combinations were observed (blade no., wind speed, TSR, turbine speed, torque coefficient, predicted clearance, clearance error) as [(40, 7.5, 0.019, 8.976, 32.309, 0.5, 2.04; (40, 8.5, 0.032, 16.999, 15.883, 0.5, 2.04; (48, 8.5, 0.009, 46.152, 62.836, 3.00, 1.69)], respectively, and the highest power coefficient for the corresponding combination was indicated as 0.499, 0.493, and 0.485. The predicted values were very close to the actual value for individual inputs, which also indicates the accuracy of the proposed model as a viable tool for evaluating SWTs' performance. Some positive combinations of this study are found to be valuable in addition to the ideal estimating method with minimal error.

Table 13(a) depicts the total computational time recorded for Storti's model and the proposed model. The proposed ANN-based virtual clone model required 5.23 min; in contrast, Storti's model needed 27.17 min for the total execution. The proposed model's execution period is 5.2 times faster (i.e., a saving of 81%). Finally, the proposed models' performance was validated with the experimental data from Case II, which showed over 98% fidelity. The predicted power coefficient value and experimental value matched together as shown in Table 13(b), which also confirms the model's best adaptation to the experimental dataset. The close agreement indicates the reliability and suitability of the developed ANN-based virtual clone model for wind turbine blade design and performance prediction.

5. Implications for industry of the developed virtual clone model

The developed ANN-based virtual clone model can ensure its viability in a variety of potential domains, including wind turbine technology, aircraft aerodynamic shape analysis, road vehicle aerodynamic performance optimization, and others [89,90]. Aerodynamic shape and performance optimization starts with a sampled parametric dataset, where the initial step is to design the 3D model of the intended system or device and then develop the critical simulation environment (CFD, FFT, etc.) or incorporate the experimental setup for a certain test.

Table 13 (a)
Computational cost of Storti's model and proposed model.

Description	Storti's study	Current study
Metamodels training	1620 s	310 s
Time for Model's C_p estimation	10 s	4 s
Total time	27.17 min	5.23 min
Time saved	–	80.75%

Table 13 (b)
STW's coefficient of power from both the experimental and ANN-based virtual clone approaches with the Case II dataset.

Approach	Average Coefficient of Power (C_p)
Loganathan's experimental study	0.2421
Proposed ANN based virtual clone model	0.2351

a) Wind Turbine Technology

The majority of wind turbine designs are carried out using computational fluid dynamics modelling and/or wind tunnel testing. These conventional approaches can be replaced with the developed model for optimizing the airfoil shape, number of blades, rotor diameter, aspect ratio, overlap ratio, and other aerodynamic parameters of any specific turbine. Additionally, the proposed model can also be utilized for wind mapping and wind-speed prediction for any potential wind farms.

b) Aircraft and Road Vehicle Aerodynamic Performance Optimization

Aerodynamic shape optimization is critical for the performance of aircraft and road vehicles. The outer shape of the vehicle, aircraft, or spacecraft must be aerodynamically efficient in order to reduce aerodynamic drag. The proposed virtual clone model can be implemented to optimize the better aerodynamic structure and reduce the cost.

c) Mechanical Parts Performance Analysis

Although beyond the scope of this study, the proposed ANN-based virtual clone model can be used to analyze the parametric value for the design and development of mechanical equipment such as gears, shafts, and rotors. When using the traditional simulation method, prior investigation of the sampled variables on the developed model can speed up the efficient design and installation of the mentioned quantities.

6. Conclusions

The aim of the study was to develop an ANN-based virtual clone model to determine and optimize wind turbine performance with minimal time, cost, and effort, independent of existing computational fluid dynamics or experimental methods. An artificial neural network (ANN)-based virtual clone model has been developed, incorporating computational and experimental data available in the public domain, to achieve this objective.

The following are the major findings:

1) The developed ANN-based virtual clone model is much faster compared to existing simulation models, thereby saving computational time.
2) The validation of the proposed model against experimental data is at over 98% fidelity. The close agreement indicates the reliability and suitability of the developed ANN-based virtual clone model for wind turbine blade design and performance prediction. This will enable an alternative method of determining, optimizing, and augmenting wind turbine efficiency.
3) Using the developed ANN-based virtual clone model and the embedded linear search technique, the optimal input combination for determining the SWT's power coefficient can be identified. The power coefficient is one of the major decision-making parameters for efficient and optimized wind turbines.
4) The developed ANN-based virtual clone model has significant industrial implications as it can offer an accelerated design process not only in wind turbine design but also in other closely related fields such as aeronautical and aerospace engineering, road vehicle and high-speed train design, and wind resource mapping.
5) The efficacy of the developed ANN-based virtual clone model can be further enhanced by using larger input data sets, including complex in-situ turbulence characteristics and wind statistical information.

Author contribution statement

Abdullah Al Noman: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Wrote the paper.

Zinat Tasneem: Conceived and designed the experiments; Analyzed and interpreted the data.

Sarafat Hussain Abhi, Faisal R Badal: Analyzed and interpreted the data.

Md Rafsanzane, Md Robiul Islam, Firoz Alam: Contributed reagents, materials, analysis tools or data.

Data availability statement

The authors do not have permission to share data.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] B.R. Loganathan, Aerodynamic Study of Single Stage Multi-Blade Drag-Based Vertical Axis Wind Turbines", PhD Thesis, School of Engineering, RMIT University, 2018.
[2] Global Wind Report 2022 - Global Wind Energy Council. https://gwec.net/global-wind-report-2022/. (Accessed 25 August 2022) accessed.
[3] Z. Tasneem, A.A. Noman, S.K. Das, D.K. Saha, M.R. Islam, M.F. Ali, M.F.R. Badal, M.H. Ahamed, S.I. Muyeen, F. Alam, An analytical review on the evaluation of wind resource and wind turbine for urban application: prospect and challenges, Develop. Built Environ. 4 (2020), 100033, Nov, https://doi: 10.1016/j.dibe.2020.100033.
[4] Why is renewable energy important? - REN21. https://www.ren21.net/why-is-renewable-energy-important/. (Accessed 25 August 2022) accessed.
[5] T. Stathopoulos, H. Alrawashed, A. Al-Quraan, B. Blocken, A. Dilimulati, M. Paraschivoiu, P. Pilay, Urban wind energy: some views on potential and challenges, J. Wind Eng. Ind. Aerod. 179 (2018) 146–157, https://doi.org/10.1016/j.jweia.2018.05.018. Aug.
[6] F. Toja-Silva, A. Colmenar-Santos, M. Castro-Gil, Urban wind energy exploitation systems: behaviour under multidirectional flow conditions - opportunities and challenges, Renew. Sustain. Energy Rev. 24 (2013) 364–378, https://doi.org/10.1016/j.rser.2013.03.052. Aug. 01.
[7] B. Loganathan, I. Mustary, H. Chowdhury, F. Alam, Effect of sizing of a Savonius type vertical axis micro wind turbine, Energy Procedia 110 (2017) 555–560, https://doi.org/10.1016/j.egypro.2017.03.184.
[8] M.R. Islam, S. Mekhilef, R. Saidur, Progress and recent trends of wind energy technology, Renew. Sustain. Energy Rev. 21 (2013) 456–468, https://doi.org/10.1016/j.rser.2013.01.007. May 01.
[9] R. Saidur, N.A. Rahim, M.R. Islam, K.H. Solangi, Environmental impact of wind energy, Renew. Sustain. Energy Rev. 15 (5) (2011) 2423–2430, Jun, https://doi.org/10.1016/j.rser.2011.02.024, 01.
[10] S.S. Bhuyan, A. Biswas, Investigations on self-starting and performance characteristics of simple H and hybrid H-Savonius vertical axis wind rotors, Energy Convers. Manag. 87 (Aug. 2014) 859–867, https://doi.org/10.1016/j.enconman.2014.07.056.
[11] F. Balduzzi, A. Bianchini, E.A. Carnevale, L. Ferrari, S. Magnani, Feasibility analysis of a Darrieus vertical-axis wind turbine installation in the rooftop of a building, Appl. Energy 97 (Sep. 2012) 921–929, https://doi.org/10.1016/j.apenergy.2011.12.008.
[12] J. Chen, H. Yang, M. Yang, H. Xu, Z. Hu, A comprehensive review of the theoretical approaches for the airfoil design of lift-type vertical axis wind turbine, Renew. Sustain. Energy Rev. 51 (2015) 1709–1720, https://doi.org/10.1016/j.rser.2015.07.065. Aug. 11.
[13] M. Ghasemian, Z.N. Ashrafi, A. Sedaghat, A review on computational fluid dynamic simulation techniques for Darrieus vertical axis wind turbines, Energy Convers. Manag. 149 (2017) 87–100, https://doi.org/10.1016/j.enconman.2017.07.016. Oct. 01.
[14] B. Hand, G. Kelly, A. Cashman, Aerodynamic design and performance parameters of a lift-type vertical axis wind turbine: a comprehensive review, Renew. Sustain. Energy Rev. 139 (2021), https://doi.org/10.1016/j.rser.2020.110699, 110699, Apr. 01.
[15] P.M. Kumar, K. Sivalingam, S. Narasimalu, T.-C. Lim, S. Ramakrishna, H. Wei, A review on the evolution of darrieus vertical Axis wind turbine: small wind turbines, J. Power Energy Eng. 7 (4) (2019) 27–44, https://doi.org/10.4236/jpee.2019.74002. Apr.
[16] R.K. Reja, R. Amin, Z. Tasneem, M.F. Ali, M.R. Islam, D.K. Saha, F.R. Badal, M.H. Ahamed, S.I. Muyeen, S.K. Das, A review of the evaluation of urban wind resources: challenges and perspectives, Energy Build. 257 (2022), 111781, https://doi.org/10.1016/j.enbuild.2021.111781.
[17] R. Kumar, K. Raahemifar, A.S. Fung, A critical review of vertical axis wind turbines for urban applications, Renew. Sustain. Energy Rev. 89 (2018) 281–291, https://doi.org/10.1016/j.rser.2018.03.033. Jun. 01.
[18] A.A. Noman, Z. Tasneem, M.F. Sahed, S.M. Muyeen, S.K. Das, F. Alam, Towards next generation Savonius wind turbine: artificial intelligence in blade design trends and framework, Renew. Sustain. Energy Rev. 168 (2022), 112531, https://doi.org/10.1016/j.rser.2022.112531.
[19] I. Antonopoulos, V. Robu, B. Couraud, D. Kirli, S. Norbu, A. Kiprakis, D. Flynn, S.E. Gonzalez, S. Wattam, Artificial intelligence and machine learning approaches to energy demand-side response: a systematic review, Renew. Sustain. Energy Rev. 130 (2020), https://doi.org/10.1016/j.rser.2020.109899, 109899, Sep.
[20] C.M. Corcoran, G.A. Cecchi, Using Language processing and speech analysis for the identification of psychosis and other disorders, Biol. Psychiatr.: Cogn. Neurosci. Neuroimag. 5 (8) (2020) 770–779, https://doi.org/10.1016/j.bpsc.2020.06.004. Aug. 01.
[21] C.W. Transform, C. Neural, A novel approach to detect cardiac arrhythmia based on continuous wavelet transform and convolutional neural network, MIST Int. J. Sci. Tech. 10 (2022) 37–41. December.
[22] A. Aboelezz, P. Makeen, H.A. Ghali, G. Elbayomi, M.M. Abdelrahman, Electric Vehicle Battery Charging Framework Using Artificial Intelligence Modeling of a Small Wind Turbine Based on Experimental Characterization, Clean Technologies and Environmental Policy, 2022, 0123456789, https://doi.org/10.1007/s10098-022-02430-x.
[23] S. Higgins, T. Stathopoulos, Application of artificial intelligence to urban wind energy, Build. Environ. 197 (2021), https://doi.org/10.1016/j.buildenv.2021.107848. April.
[24] M. Trisakti, L. Halim, B.M. Arthaya, Power Coefficient Analysis of Savionus Wind Turbine Using CFD Analysis," Proc. 2019 International Conference on Mechatronics, Robotics and System Engineering MoRSE, 2019, pp. 24–29, https://doi.org/10.1109/MoRSE48060.2019.8998703. Dec. 2019.
[25] D. Hilewit, E.A. Matida, A. Fereidooni, H.A. el Ella, F. Nitzsche, Power coefficient measurements of a novel vertical axis wind turbine, Energy Sci. Eng. 7 (6) (2019) 2373–2382, Dec, https://doi.org/10.1002/ese3.412.
[26] K.S. Jeon, J.I. Jeong, J.K. Pan, K.W. Ryu, Effects of end plates with various shapes and sizes on helical Savonius wind turbines, Renew. Energy 79 (1) (Jul. 2015) 167–176, https://doi.org/10.1016/j.renene.2014.11.035.
[27] K.H. Wong, W.T. Chong, N.L. Sukiman, S.C. Poh, Y.C. Shiah, C.T. Wang, Performance enhancements on vertical axis wind turbines using flow augmentation systems: a review, Renew. Sustain. Energy Rev. 73 (2017) 904–921, https://doi.org/10.1016/j.rser.2017.01.160. Jun. 01.
[28] U.K. Saha, S. Thotla, D. Maity, Optimum design configuration of Savonius rotor through wind tunnel experiments, J. Wind Eng. Ind. Aerod. 96 (8–9) (2008) 1359–1375, Aug, https://doi.org/10.1016/j.jweia.2008.03.00.
[29] J.L. Menet, A double-step Savonius rotor for local production of electricity: a design study, Renew. Energy 29 (11) (2004) 1843–1862, Sep, https://doi.org/10.1016/j.renene.2004.02.011.
[30] A. Damak, Z. Driss, M.S. Abid, Experimental investigation of helical Savonius rotor with a twist of 180, Renew. Energy 52 (Apr. 2013) 136–142, https://doi.org/10.1016/j.renene.2012.10.043.
[31] M. Zemamou, M. Aggour, A. Toumi, Review of Savonius Wind Turbine Design and Performance, vol. 141, Energy Procedia, Dec. 2017, pp. 383–388, https://doi.org/10.1016/j.egypro.2017.11.047.
[32] A. Dewan, A. Gautam, R. Goyal, Savonius wind turbines: a review of recent advances in design and performance enhancements, Mater. Today Proc. (May 2021), https://doi.org/10.1016/j.matpr.2021.05.20.
[33] S. Fanel Dorel, G. Adrian Mihai, D. Nicusor, Review of specific performance parameters of vertical wind turbine rotors based on the SAVONIUS type, Energies 14 (7) (2021), 1962, Apr, https://doi: 10.3390/en14071962.
[34] A. Banerjee, S. Roy, P. Mukherjee, U.K. Saha, Unsteady Flow Analysis Around an Elliptic-Bladed Savonius-Style Wind Turbine, Feb. 2014, https://doi.org/10.1115/GTINDIA2014-8141.
[35] M. Tartuferi, V. D'Alessandro, S. Montelpare, R. Ricci, Enhancement of savonius wind rotor aerodynamic performance: a computational study of new blade shapes and curtain systems, Energy 79 (Jan. 2015) 371–384, https://doi.org/10.1016/j.energy.2014.11.023. C.

[36] S. Sharma, R.K. Sharma, Performance improvement of Savonius rotor using multiple quarter blades – a CFD investigation, Energy Convers. Manag. 127 (2016) 43–54, https://doi.org/10.1016/j.enconman.2016.08.087. Nov.

[37] S. Sharma, R.K. Sharma, CFD Investigation to Quantify the Effect of Layered Multiple Miniature Blades on the Performance of Savonius Rotor, vol. 144, Energy Conversion and Management, Jul. 2017, pp. 275–285, https://doi.org/10.1016/j.enconman.2017.04.059.

[38] B. Emmanuel, W. Jun, Numerical study of a six-bladed savonius wind turbine, J. Sol. Energy Eng. 133 (4) (2011), https://doi.org/10.1115/1.4004549. Nov.

[39] N. Alom, U.K. Saha, Four decades of research into the augmentation techniques of savonius wind turbine rotor, J. Energy Resour. Technol. 140 (5) (May 2018), https://doi.org/10.1115/1.4038785.

[40] C. Harsito, D.D.D.P. Tjahjana, B. Kristiawan, Savonius turbine performance with slotted blades, AIP Conf. Proceed. 2217 (1) (2020), 030071, https://doi.org/10.1063/5.0000797.

[41] S. Roy, P. Mukherjee, U.K. Saha, Aerodynamic Performance Evaluation of a Novel Savonius-Style Wind Turbine under an Oriented Jet," *Gas Turbine India Conference*, Feb. 2014, https://doi.org/10.1115/GTINDIA2014-8152.

[42] A.J. Alexander, B.P. Holownia, Wind tunnel tests on a savonius rotor, J. Wind Eng. Ind. Aerod. 3 (4) (Jan. 1978) 343–351, https://doi.org/10.1016/0167-6105(78)90037-5.

[43] S. Acarer, Ç. Uyulan, Z.H. Karadeniz, Optimization of radial inflow wind turbines for urban wind energy harvesting, Energy 202 (2020), https://doi.org/10.1016/j.energy.2020.117772, 117772, Jul.

[44] T. Ogawa, H. Yoshida, Y. Yokota, Development of rotational speed control systems for a savonius-type wind turbine, J. Fluid. Eng. Trans. ASME 111 (1) (Mar. 1989) 53–58, https://doi.org/10.1115/1.3243598.

[45] A.S. Grinspan, U.K. Saha, P. Mahanta, Experimental Investigation of Twisted Bladed Savonius Wind Turbine Rotor, Apr. 2004.

[46] K. Kacprzak, G. Liskiewicz, K. Sobczak, Numerical investigation of conventional and modified Savonius wind turbines, Renew. Energy 60 (Dec. 2013) 578–585, https://doi.org/10.1016/j.renene.2013.06.009.

[47] L. Song, Z.X. Yang, R.T. Deng, X.G. Yang, Performance and Structure Optimization for a New Type of Vertical axis Wind Turbine," *International Conference on Advanced Mechatronic Systems*, ICAMechS, 2013, pp. 687–692, https://doi.org/10.1109/ICAMechS.2013.6681730.

[48] G.G. Muscolo, R. Molfino, From Savonius to Bronzinus: a comparison among vertical wind turbines, Energy Procedia 50 (2014) 10–18, https://doi.org/10.1016/j.egypro.2014.06.002.

[49] N. Alom, S.C. Kolaparthi, S.C. Gadde, U.K. Saha, Aerodynamic design optimization of elliptical-bladed savonius-style wind turbine by numerical simulations, in: Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE, 6, Oct. 2016, https://doi.org/10.1115/OMAE2016-55095.

[50] M. Mari, M. Venturini, A. Beyene, A novel geometry for vertical axis wind turbines based on the savonius concept, J. Energy Res. Techn. Trans. ASME 139 (6) (2017), https://doi.org/10.1115/1.4036964. Nov.

[51] Y. Kurniawan, D. Danardono, D. Prija Tjahjana, B. Santoso, Experimental study of savonius wind turbine performance with blade layer addition, J. Adv. Res. Fluid Mech. Ther. Sci. 69 (2020) 23–33, https://doi.org/10.37934/arfmts.69.1.2333.

[52] B.A. Storti, J.J. Dorella, N.D. Roman, I. Peralta, A.E. Albanesi, Improving the efficiency of a Savonius wind turbine by designing a set of deflector plates with a metamodel-based optimization approach, Energy 186 (2019), https://doi.org/10.1016/j.energy.2019.07.144, 115814, Nov.

[53] C.M. Chan, H.L. Bai, D.Q. He, Blade shape optimization of the Savonius wind turbine using a genetic algorithm, Appl. Energy 213 (Mar. 2018) 148–157, https://doi.org/10.1016/j.apenergy.2018.01.029.

[54] M. Masdari, M. Tahani, M.H. Naderi, N. Babayan, Optimization of airfoil Based Savonius wind turbine using coupled discrete vortex method and salp swarm algorithm, J. Clean. Prod. 222 (Jun. 2019) 47–56, https://doi.org/10.1016/j.jclepro.2019.02.237.

[55] M.J. Chern, D. Goytom Tewolde, C.C. Kao, N. Vaziri, Vertical-Axis wind turbine blade-shape optimization using a genetic algorithm and direct-forcing immersed boundary method, J. Energy Eng. 147 (2) (2021), https://doi.org/10.1061/(asce)ey.1943-7897.0000741, 04020091, Apr.

[56] W.A. El-Askary, M.H. Nasef, A.A. AbdEL-hamid, H.E. Gad, Harvesting wind energy for improving performance of savonius rotor, J. Wind Eng. Ind. Aerod. 139 (Apr. 2015) 8–15, https://doi.org/10.1016/j.jweia.2015.01.003.

[57] M.B. Salleh, N.M. Kamaruddin, Z. Mohamed-Kassim, The effects of deflector longitudinal position and height on the power performance of a conventional Savonius turbine, Energy Convers. Manag. 226 (2020), https://doi.org/10.1016/j.enconman.2020.113584, 113584, Dec.

[58] S.M. Morcos, M.G. Khalafallah, H.A. Heikel, S.M. Morcos, M.G. Khalafallah, H.A. Heikel, The effect of shielding on the aerodynamic performance of Savonius wind turbines, IECE 2 (1981) 2037–2040.

[59] T. Ogawa, H. Yoshida, The effects of a deflecting plate and rotor end plates on performances of savonius-type wind turbine, Bullet. JSME 29 (253) (1986) 2115–2121, https://doi.org/10.1299/jsme1958.29.2115.

[60] P. Reupke, S.D. Probert, Slatted-blade savonius wind-rotors, Appl. Energy 40 (1) (Jan. 1991) 65–75, https://doi.org/10.1016/0306-2619(91)90051-X.

[61] B.D. Altan, M. Atilgan, The use of a curtain design to increase the performance level of a Savonius wind rotors, Renew. Energy 35 (4) (Apr. 2010) 821–829, https://doi.org/10.1016/j.renene.2009.08.025.

[62] M.H. Mohamed, G. Janiga, E. Pap, D. Thèvenin, Optimization of Savonius turbines using an obstacle shielding the returning blade, Renew. Energy 35 (11) (2010) 2618–2626, Nov, https://doi.org/10.1016/j.renene.2010.04.007.

[63] D. Danardono Dwi Prija Tjahjana, et al., Study on performance improvement of the savonius wind turbine for urban power system with omni-directional guide vane (ODGV), J. Adv. Res. Fluid Mech. Ther. Sci. 55 (2019) 126–135.

[64] T. Brahimi, Using artificial intelligence to predict wind speed for energy application in Saudi arabia, Energies 12 (24) (2019), https://doi.org/10.3390/en12244669, 4669, Dec.

[65] J. Jayabalan, D. Yildirim, D. Kim, P. Samui, Design Optimization of a Wind Turbine Using Artificial Intelligence, Mathematical Concepts and Applications in Mechanical Engineering and Mechatronics, 2017, pp. 38–66, https://doi.org/10.4018/978-1-5225-1639-2.ch003.

[66] N. Sharma, R. Sharma, N. Jindal, Machine learning and deep learning applications-A vision, Global Trans. Proceed. 2 (1) (Jun. 2021) 24–28, https://doi.org/10.1016/j.gltp.2021.01.004.

[67] Y. Lecun, Y. Bengio, G. Hinton, Deep Learning," *Nature*, vol. 521, Nature Publishing Group, 2015, p. 7553, https://doi.org/10.1038/nature14539, 436–444, May 27.

[68] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G.V. D. Driessche, T. Graepel, D. Hassabis, Mastering the game of Go without human knowledge, Nature 550 (7676) (Oct. 2017) 354–359, https://doi.org/10.1038/nature24270.

[69] A. Biswas, S. Sarkar, R. Gupta, Application of artificial neural network for performance evaluation of vertical axis wind turbine rotor, Int. J. Ambient Energy 37 (2) (2016) 209–218, https://doi.org/10.1080/01430750.2014.915889.

[70] S. Teksin, N. Azginoglu, S.O. Akansu, Structure estimation of vertical axis wind turbine using artificial neural network, Alex. Eng. J. 61 (1) (2022) 305–314, https://doi.org/10.1016/j.aej.2021.05.002.

[71] M.D. Manshadi, M. Ghassemi, S.M. Mousavi, A.H. Mosavi, L. Kovacs, Predicting the parameters of vortex bladeless wind turbine using deep learning method of long short-term memory, Energies 14 (2021), https://doi.org/10.3390/en14164867. *Page 4867*, vol. 14, no. 16, p. 4867, Aug.

[72] H.A.H.R. Aladwani, M.K.A. Ariffin, F. Mustapha, A supervised machine-learning method for optimizing the automatic transmission system of wind turbines, Eng. Solid Mech. 10 (1) (2022) 35–56, https://doi.org/10.5267/J.ESM.2021.11.001.

[73] B. Wilson, S. Wakes, M. Mayo, Surrogate modeling a computational fluid dynamics-based wind turbine wake simulation using machine learning, in: IEEE Symposium Series on Computational Intelligence, 2017, https://doi.org/10.1109/SSCI.2017.8280844. *SSCI 2017)*, vol. 2018-Janua, pp. 1–8, 2018.

[74] M. Mohammadi, M. Lakestani, M.H. Mohamed, Intelligent parameter optimization of savonius rotor using artificial neural network and genetic algorithm, Energy 143 (Jan. 2018) 56–68, https://doi.org/10.1016/j.energy.2017.10.121.

[75] T. M. Ahmed, "Prediction of Aerodynamic Characteristics of Savonius Wind Turbine Using Artificial Neural Network and Fourier Series." Thesis submitted to the Graduate School of Applied Science of Near East University, Nicosia, Cyprus.

[76] P.A. Kulkarni, A.S. Dhoble, P.M. Padole, Deep neural network-based wind speed forecasting and fatigue analysis of a large composite wind turbine blade, Proc. IME C J. Mech. Eng. Sci. 233 (8) (2019) 2794–2812, https://doi.org/10.1177/0954406218797972.
[77] U.H. Rathod, V. Kulkarni, U.K. Saha, On the application of machine learning in savonius wind turbine technology: an estimation of turbine performance using artificial neural network and genetic expression programming, J. Energy Resour. Technol. 144 (6) (2022), https://doi.org/10.1115/1.4051736.
[78] M.A. Moreno-Armendáriz, E. Ibarra-Ontiveros, H. Calvo, C.A. Duchanoy, Integrated surrogate optimization of a vertical Axis wind turbine, Energies 15 (1) (Dec. 2021) 233, https://doi.org/10.3390/en15010233. *Page 233*, vol. 15.
[79] S.A. Al-Shammari, A.H. Karamallah, S. Aljabair, Blade shape optimization of savonius wind turbine at low wind energy by artificial neural network, IOP Conf. Ser. Mater. Sci. Eng. 881 (1) (2020), 012154, https://doi.org/10.1088/1757-899X/881/1/012154. Aug.
[80] W.H. Chen, J.S. Wang, M.H. Chang, A.T. Hoang, S.S. Lam, E.E. Kwon, V. Ashokkumar, Optimization of a vertical axis wind turbine with a deflector under unsteady wind conditions via Taguchi and neural network applications, Energy Convers. Manag. 254 (Feb. 2022), https://doi.org/10.1016/j.enconman.2022.115209. Available:.
[81] J. Joseph, P. Pant, S.N. Omkar, Optimisation framework for distinctive vertical axis wind turbine blade generation using hybrid multi-objective genetic algorithms and deep neural networks, in: AIAA AVIATION 2020 FORUM, 1 PartF, 2020, pp. 1–31, https://doi.org/10.2514/6.2020-3119.
[82] W. Tian, B. Song, J. VanZwieten, P. Pyakurel, Computational fluid dynamics prediction of a modified savonius wind turbine with novel blade shapes, Energies 8 (8) (2015) 7915–7929, Jul, https://doi.org/10.3390/en8087915.
[83] Introduction to ANN (artificial neural networks) | set 3 (hybrid systems) - GeeksforGeeks. https://www.geeksforgeeks.org/introduction-ann-artificial-neural-networks-set-3-hybrid-systems/ accessed Jan. 30, 2023).
[84] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity (reprinted from bulletin of mathematical biophysics, vol 5, pg 115-133, Bull. Math. Biol. 52 (1–2) (1990) 99–115, 1943.
[85] Neural Networks and Learning Machines, 3/e - Simon Haykin - Google Books, 2022 accessed aug. 05.
[86] S. Trenn, Multilayer perceptrons: approximation order and necessary number of hidden units, IEEE Trans. Neural Network. 19 (5) (2008) 836–844, https://doi.org/10.1109/TNN.2007.912306.
[87] S. Deb, A novel robust r-squared measure and its applications in linear regression, in: International Conference on Computational Intelligence in Information System, 532, 2017, pp. 131–142.
[88] A. Botchkarev, Evaluating performance of regression machine learning models using multiple error metrics in azure machine learning studio," social science research network, (SSRN) e-Journal (May. 2018), https://doi.org/10.2139/ssrn.3177507.
[89] J. Li, X. Du, J.R.R.A. Martins, Machine learning in aerodynamic shape optimization, Prog. Aero. Sci. 134 (2022), 100849, https://doi.org/10.1016/j.paerosci.2022.100849. July.
[90] X. Zhang, F. Xie, T. Ji, Z. Zhu, Y. Zheng, Multi-fidelity deep neural network surrogate model for aerodynamic shape optimization, Comput. Methods Appl. Mech. Eng. 373 (Jan) (2021), https://doi.org/10.1016/j.cma.2020.113485.