IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 17, 2021, accepted August 31, 2021, date of publication September 10, 2021, date of current version September 21, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3111667

Wind Speed Forecasting Using the Stationary Wavelet Transform and Quaternion Adaptive-Gradient Methods

LYES SAAD SAOUD[1], HASAN AL-MARZOUQI[1], (Senior Member, IEEE), AND MOHAMED DERICHE[2,3], (Senior Member, IEEE)

[1]Department of Electrical and Computer Engineering, Khalifa University, Abu Dhabi, United Arab Emirates
[2]Artificial Intelligence Research Center (AIRC), College of Engineering and IT, Ajman University, Ajman, United Arab Emirates
[3]Electrical Engineering Department, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia

Corresponding author: Lyes Saad Saoud (lyes.saoud@ku.ac.ae)

This work was supported by the Information and Communication Technology (ICT), Telecommunications Regulatory Authority (TRA), Abu Dhabi, United Arab Emirates.

ABSTRACT Accurate wind speed forecasting is a fundamental requirement for advanced and economically viable large-scale wind power integration. The hybridization of the quaternion-valued neural networks and stationary wavelet transform has not been proposed before. In this paper, we propose a novel wind-speed forecasting model that combines the stationary wavelet transform with quaternion-valued neural networks. The proposed model represents wavelet subbands in quaternion vectors, which avoid separating the naturally correlated subbands. The model consists of three main steps. First, the wind speed signal is decomposed using the stationary wavelet transform into sublevels. Second, a quaternion-valued neural network is used to forecast wind speed components in the stationary wavelet domain. Finally, the inverse stationary wavelet transform is applied to estimate the predicted wind speed. In addition, a softplus quaternion variant of the RMSProp learning algorithm is developed and used to improve the performance and convergence speed of the proposed model. The proposed model is tested on wind speed data collected from different sites in China and the United States, and the results demonstrate that it consistently outperforms similar models. In the meteorological terminal aviation routine (METAR) dataset experiment, the proposed wind speed forecasting model reduces the mean absolute error, and root mean squared error of predicted wind speed values by 26.5% and 33%, respectively, in comparison to several existing approaches.

INDEX TERMS Wind speed forecasting, stationary wavelet transform, quaternion valued neural network, RMSProp learning algorithm.

I. INTRODUCTION

Renewable energy plays an increasingly imperative role in the global energy market [1]. Among renewable energy resources, wind energy has attracted much attention due to its mature technology, low cost, and climate change impacts regarding reducing environmental pollution. Currently, wind power is one of the fastest-growing renewable energy technologies, and according to the global wind energy council report [2], 2019 witnessed new wind power installations surpassing 60 GW globally, representing a 19% increase compared with 2018, bringing the total installed capacity to 650 GW, a rise of 10 % compared with the preceding year. The world's top market in new installations in 2019 was China, with the installation of more than 2.3 GW offshore wind in a single year. Generally, accurate wind speed forecasting is a precondition for constituting an advanced and economically viable control strategy in a modern power system, e.g., model predictive control [3]. In addition, forecasting errors can significantly affect the cost of balancing the power system [4]. Therefore, accurate short-term wind speed prediction is essential for reducing wind farm operations and maintenance costs [5]. The generated power will depend on wind speed and the design of the turbine. Forecasting wind speed instead of generated power avoids dependencies on generator design.

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Asif.

In recent decades, a large number of forecasting models for wind speed have been developed and used to compute estimates of wind energy. Wind speed prediction models can be divided into three categories: physical models, statistical models, and machine learning models. Physical models consider weather conditions such as wind speed time series, air pressure, humidity, and temperature to obtain wind speed forecasts. These numerical weather prediction models (NWPs) divide the atmosphere into 3D cubes and solve weather parameter equations for each atmospheric variable at each grid point. The weather research and forecasting model (WRF) [6], the COSMO model [7] and MM5 [8] are examples of prediction models in this category.

In general, NWP models are not suitable for short-term wind speeds because of their complex calculation processes and poor performance [9]–[11].

Statistical models extract rules that govern the relationship between sequences of previous measurements and use these rules in prediction. When compared with physical methods, these methods can provide more accurate results for short-term wind speed forecasting. Examples of methods used in this category include the autoregressive integrated moving average (ARIMA) [12], Markov chains [13], and Kalman filtering [14].

Machine learning models are the third class of forecasting algorithms. They have been widely applied in predicting wind speed with good learning ability and nonlinear mapping ability [15]. Examples of these methods include neural networks, fuzzy-based systems, and decision trees. For instance, Zhou *et al.* proposed a long-short-term memory (LSTM) based lower and upper bound estimation model to construct the prediction intervals of wind power [16]. Kisvari *et al.* [17] presented a predictive model that uses gated recurrent units (GRU) and 12 features for wind speed forecasting. Examples of these features include wind speeds at four different heights, generator temperature, and gearbox temperature. Extreme gradient boosting (XGBoost), which is a form of gradient boosting decision tree, was used in [18] for wind speed forecasting.

Hybrid models can be formed by combining methods from the preceding categories. The main reason for integrating forecasting models is that a single forecasting method often has some essential weaknesses that lead to poor prediction results that do not adapt to the complex and changeable environment [19]. For example, a hybrid deep neural network model based on a stacked autoencoder and an LSTM network to forecast wind speed was proposed [20]. The presented hybrid model reduced the mean absolute error in prediction by 13% compared with a nonhybrid version of the model. In [21], a short-term multimodal wind-speed prediction framework was proposed based on rough artificial neural networks and stacked denoising autoencoders. Khosravi *et al.* [22] combined an adaptive neuro-fuzzy inference system with a particle swarm optimization algorithm to predict wind speed, wind direction, and a wind turbine's output power. The combined model showed considerable improvements in prediction accuracy compared with models that use either adaptive inference systems or particle swarm optimization. Statistical and neural network-based approaches were combined to predict hourly wind speed data [23]. The hybridization of ant colony optimization and particle swarm optimization for forecasting wind speed was proposed in [24]. This hybridization approach showed better wind speed forecasting results than other nonhybrid models.

Signal decomposition algorithms, such as the wavelet transform and empirical mode decomposition, provide efficient means for adapting the learned forecasting models to the different components of the original wind speed series. These methods were used successfully to enhance the performance of several wind speed forecasting methods. For example, Hu *et al.* [15] proposed a prediction model based on variational mode decomposition and an improved and echo state network. The proposed algorithm outperformed nine comparative models in four wind speed datasets. Variational mode decomposition (VMD) was used in [25] to decompose wind speed data into different subbands. LSTM units were used to predict the main trend, and a kernel density estimation method was used to perform predictions on the residual part. A similar approach was offered in [26], where VMD decomposition signals were used as input to convolutional LSTM prediction units.

The wavelet transform and packet decomposition are often combined with other AI models for wind power and wind speed forecasts [27]. For example, in [28], wavelet soft threshold denoising and gated recurrent units were combined to forecast wind speed. In the proposed model, denoising by wavelet soft threshold was used to filter the noisy samples from the wind signal, and a gated recurrent unit was used as a forecaster. Liu *et al.* [29] used the empirical wavelet transform to decompose raw wind speed data into several sublayers. The signal's low-frequency components were forecasted using a long short-term memory neural network, and an Elman neural network was used to predict high-frequency components. Aasim *et al.* [30] combined the wavelet transform with an autoregressive integrated moving average (ARIMA) model to forecast wind speed data. The wavelet representation was combined with adapted LSTM units to predict wind speed in [31].

The existing hybrid models based on the wavelet decomposition technique and neural networks treat each wavelet subband individually. However, significant correlations exist between these subbands [32], and improved forecasting results can be achieved by developing algorithms that exploit these dependencies between wavelet subbands.

To address this limitation, we propose using quaternion vectors to represent wavelet subbands. A quaternion valued neural network (QVNN) is used in this paper to model the relationship between quaternion inputs and outputs representing previous and forecasted wind speed values. QVNNs have quaternion inputs and outputs and use quaternion weights and bias parameters. Representing wavelet subbands as quaternion vectors forms a unifying representation that avoids

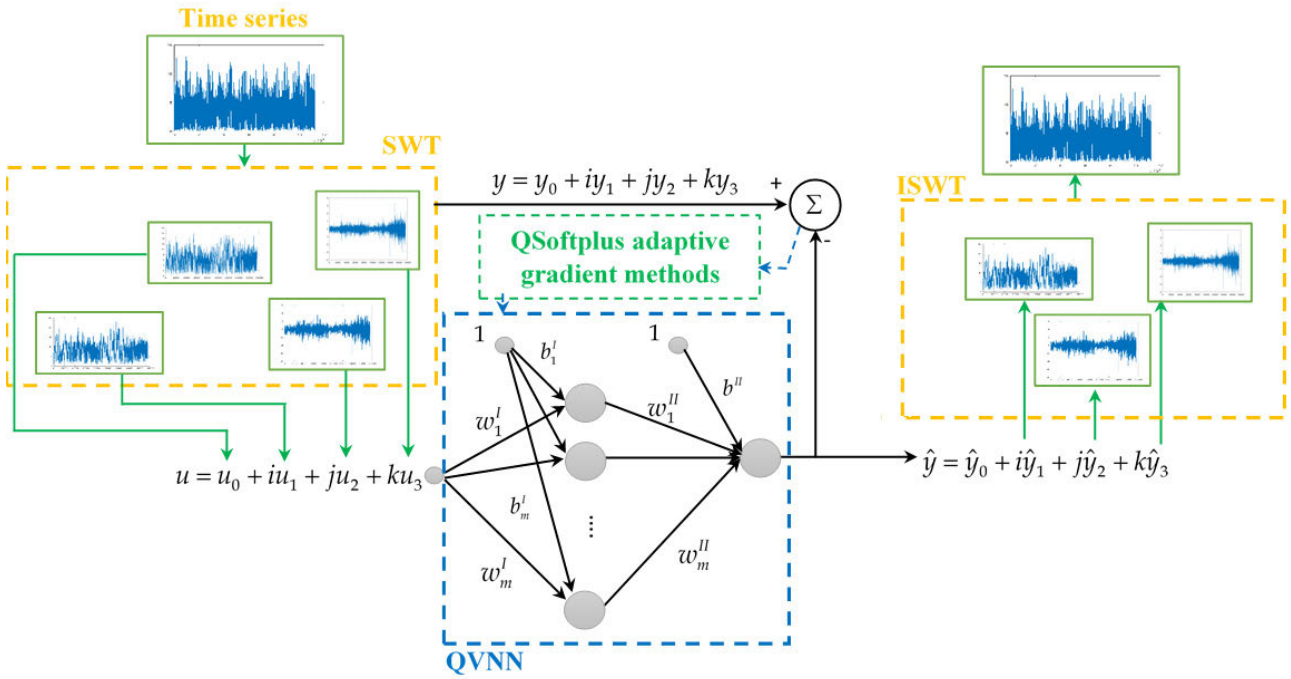

FIGURE 1. The proposed architecture. A quaternion representation of stationary wavelet subbands is used to train a quaternion-valued neural network. The predicted wind speed is computed by using the inverse stationary wavelet transform.

separating the naturally correlated sequences. QVNNs have achieved improvements in several tasks in image, speech, and signal processing [33]. They were recently used for time-series forecasting [34], where they achieved performance levels surpassing real-valued neural networks.

In the proposed system, wind speed data are first decomposed into wavelet levels using the stationary wavelet transform (Fig. 1). The resulting coefficients are used to train the QVNN to predict the wavelet subbands representing the forecasted value. The inverse stationary wavelet transform is then used to reconstruct the predicted wind speed. The adaptive quaternion learning rate algorithm is developed and used to enhance the performance of the model. Finally, the performance and convergence speed are improved by using a softplus function within the RMSProp based optimizer [35]. The introduction of the softplus function was shown to calibrate the learning rate and lead to improvements in convergence speed.

The major contributions of this paper are as follows:

- We propose a novel wind-speed forecasting model that combines the stationary wavelet transform (SWT) and quaternion-valued neural networks (QVNN). To the best of our knowledge, the hybridization between the QVNN and SWT has not been proposed before in the literature. This hybridization provides a compact and unifying representation of the different wavelet subbands.
- We propose a quaternion version of RMSProp, the adaptive learning rate algorithm, to improve accuracy and convergence speed of the proposed quaternion neural network model.
- We enhance the developed quaternion RMSProp algorithm with a softplus function to further improve the performance and convergence speed of the proposed model. In addition, the proposed softplus function prevents the forecasting model from overfitting.

The developed wind speed forecasting method is tested on wind speed data collected from different sites in China and the United States. The results demonstrate that the developed model outperforms several widely used and recently proposed wind speed prediction models.

II. PROPOSED METHODOLOGY

This section describes the three main components used in the presented model: the stationary wavelet transform, the QVNN, and the quaternion soft plus RMSProp learning algorithm.

A. THE STATIONARY WAVELET TRANSFORM

The stationary wavelet transform (SWT) [36] is designed to solve the shift-invariance issue in the discrete wavelet transform. The wavelet transform decomposes the source signal into different levels; the resulting subsignals are of a length that is equal to 1/2 of the approximation signal at the preceding wavelet level. The SWT removes the downsampling operator from the usual implementation of the DWT. The resulting subsignals in the SWT have the same length as the source signal, this is a desired property for the proposed wind speed forecasting model because this equality in length in wavelet subbands allows the formation of a quaternion vector representing the four different wavelet components of the signals at each time step.

This paper develops a quaternion-based forecasting system where each input signal is represented by a quaternion vector containing information from the four different wavelet subbands (Fig. 1). Similar to the DWT, the SWT decomposes the input series into sets of low and high-frequency coefficients called approximation and detail coefficients; however, the output signal is nondecimated (i.e., downsampled). The approximation components present the general trend of the time series, whereas the detail coefficients describe the small variations in the series (i.e., high-frequency components). The decomposition can be demonstrated as a dyadic tree [37].

Fig. 2 shows an example of a two-level decomposition based on the SWT. For a given signal $u(t)$, the SWT decomposes it into two coefficients: approximation coefficients $A_1(t)$ and detail coefficients $D_1(t)$. These coefficients represent the convolution results produced by low- and high-pass filters. This decomposition process is repeated using approximation coefficients as input in each subsequent decomposition level.

B. THE QVNN AND THE QUATERNION SOFTPLUS RMSPROP LEARNING ALGORITHM

The proposed wind speed forecasting system uses a QVNN with three layers. The input layer receives the quaternion-valued form of the two-level SWT approximation and detail coefficients $\left[A_1(t)\ D_1(t)\ A_2(t)\ D_2(t)\right]$.

The hidden layer has m neurons, and the output layer with one neuron produces a one-step forecast of the SWT approximation and detail coefficient levels (Fig. 1). The inverse SWT is used to compute the predicted time series by combining the three predicted coefficients, i.e., $A_2(t)$, $D_1(t)$, and $D_2(t)$.

The QVNN layers are connected with quaternion-valued weights w^I and w^{II}. The hidden and output layers have quaternion-valued biases b^I and b^{II}.

The predicted quaternion QVNN output can be computed as:

$$\hat{y}(k+1) = \varphi\left(\mathrm{Re}\left[\tilde{y}\right]\right) + i\,\varphi\left(\mathrm{Im}_i\left[\tilde{y}\right]\right) + j\,\varphi\left(\mathrm{Im}_j\left[\tilde{y}\right]\right) + k\,\varphi\left(\mathrm{Im}_k\left[\tilde{y}\right]\right) \quad (1)$$

where Re is the real part of the predicted vector $\hat{y}$, Im_i, Im_j and Im_k are the imaginary parts in the i, j, and k complex dimensions of the quaternion vector, respectively; and φ is the nonlinear sigmoid function given by the following equation:

$$\varphi(\cdot) = \frac{1}{1+e^{-(\cdot)}} \quad (2)$$

The quaternion vector $\tilde{y}$ is computed using

$$\tilde{y} = \sum_{p=1}^{m} w_p^{II} h_p + b^{II}, \quad (3)$$

where $p = 1, \ldots, m$ and h_p is the p^{th} hidden neuron's output given as follows

$$h_p = Q\mathrm{ReLU}\left(\tilde{h}_p\right) = \mathrm{ReLU}\left(\mathrm{Re}\left[\tilde{h}_p\right]\right) + i\,\mathrm{ReLU}\left(\mathrm{Im}_i\left[\tilde{h}_p\right]\right) + j\,\mathrm{ReLU}\left(\mathrm{Im}_j\left[\tilde{h}_p\right]\right) + k\,\mathrm{ReLU}\left(\mathrm{Im}_k\left[\tilde{h}_p\right]\right) \quad (4)$$

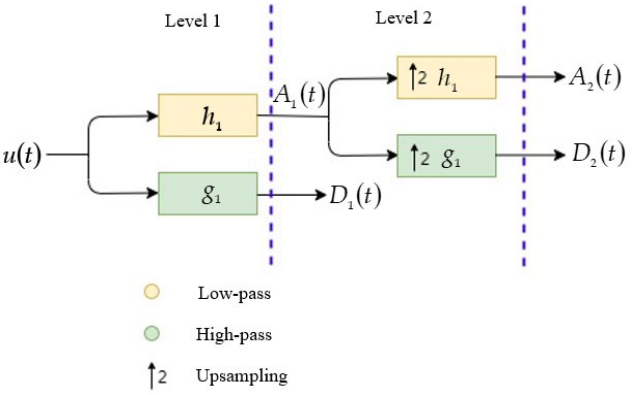

FIGURE 2. Two decomposition levels obtained using the SWT. Two approximations and the two details coefficients are used to construct a quaternion vector $u = u_1 + iu_2 + ju_3 + ku_4$, where $u_1 = A_1(t)$, $u_2 = A_2(t)$, $u_3 = D_1(t)$, and $u_4 = D_2(t)$.

where $\tilde{h}_p$ is given by

$$\tilde{h}_p = w_p^I u + b_p^I, \quad (5)$$

where u is the input which contains the quaternion-valued representation of the two level SWT approximation and detail coefficients.

The objective of this model is to find the network's optimal weights and bias parameters that minimize the sum-squared error at the output layer, which can be written as

$$E = \frac{1}{2} e^H e = \frac{1}{2}\sum_{l=1}^{N} e_l e_l^* = \frac{1}{2}\sum_{l=1}^{N} E_l \quad (6)$$

$$E_l = e_l e_l^* = |e_l|^2 \quad (7)$$

The superscript '*' represents the conjugate operator, and H is the Hermitian operator.

$$e_l = y(k+1) - \hat{y}(k+1) = \mathrm{Re}\left[e_l\right] + i\mathrm{Im}_i\left[e_l\right] + j\mathrm{Im}_j\left[e_l\right] + k\mathrm{Im}_k\left[e_l\right] \quad (8)$$

where e_l is the lth error between the l^{th} desired output y and the lth estimated output $\hat{y}$, $l = 1, \ldots N$.

We develop a quaternion version of the RMSProp learning algorithm to train the proposed QVNN. We also equip this algorithm with a quaternion softplus function (Algorithm 1) to accelerate the convergence rate [35]. $v_\theta(k)$ is the second-order quaternion momentum calculated as a combination of previous and current squared stochastic gradients.

The stochastic gradient of the QVNN is computed using the following equations (see [34], [38]–[40] for further details).

To compute the gradient of the bias,

$$b^{II} = \mathrm{Re}\left[b^{II}\right] + i\mathrm{Im}_i\left[b^{II}\right] + j\mathrm{Im}_j\left[b^{II}\right] + k\mathrm{Im}_k\left[b^{II}\right],$$

we have:

$$\nabla_{b^{II}} E = \frac{\partial E}{\partial \mathrm{Re}\left[b^{II}\right]} + i\frac{\partial E}{\partial \mathrm{Im}_i\left[b^{II}\right]} + j\frac{\partial E}{\partial \mathrm{Im}_j\left[b^{II}\right]} + k\frac{\partial E}{\partial \mathrm{Im}_k\left[b^{II}\right]} \quad (9)$$

Algorithm 1 Quaternion SRMSProp

Inputs: $\theta, u, y(k+1) \in Q$, learning rate $\eta = 0.001$, Parameters $\beta_2 = 0.999$, $\beta = 50$,
Initialize: $m_\theta(0) = 0, \widetilde{v}_\theta(0) = 0$,
for $k = 1$ to T **do**
  Compute the stochastic gradient of all weights and biases
  $\widetilde{v}_\theta(k) = \beta_2 \widetilde{v}_\theta(k-1) + (1-\beta_2)\,(\nabla_\theta E)^2$
  $v_\theta(k) = \max\,(v_\theta(k-1), \widetilde{v}_\theta(k))$
  $\theta\,(k+1) = \theta\,(k) - \frac{\eta}{Qsoftplus(\sqrt{v_\theta(k)})} \times \nabla_\theta E$
end for

$$\nabla_{b^{II}} E = -\left\{\mathrm{Re}\,[e]\,\left(1-\mathrm{Re}\left[\hat{y}\right]\right)\cdot \mathrm{Re}\left[\hat{y}\right] + i\mathrm{Im}_i\,[e] \times \left(1-\mathrm{Im}_i\left[\hat{y}\right]\right)\cdot \mathrm{Im}_i\left[\hat{y}\right] + j\mathrm{Im}_j\,[e]\,\left(1-\mathrm{Im}_j\left[\hat{y}\right]\right) \cdot \mathrm{Im}_j\left[\hat{y}\right] + k\mathrm{Im}_k\,[e]\,\left(1-\mathrm{Im}_k\left[\hat{y}\right]\right)\cdot \mathrm{Im}_k\left[\hat{y}\right]\right\} \quad (10)$$

For the weight

$$w^{II} = \mathrm{Re}\left[w^{II}\right] + i\mathrm{Im}_i\left[w^{II}\right] + j\mathrm{Im}_j\left[w^{II}\right] + k\mathrm{Im}_k\left[w^{II}\right],$$

we have:

$$\nabla_{w^{II}} E = \frac{\partial E}{\partial \mathrm{Re}\left[w^{II}\right]} + i\frac{\partial E}{\partial \mathrm{Im}_i\left[w^{II}\right]} + j\frac{\partial E}{\partial \mathrm{Im}_j\left[w^{II}\right]} + k\frac{\partial E}{\partial \mathrm{Im}_k\left[w^{II}\right]} \quad (11)$$

$$\nabla_{w^{II}} E = -h_s^* \cdot \nabla_{b^{II}} E \quad (12)$$

For the bias term,

$$b^I = \mathrm{Re}\left[b^I\right] + i\mathrm{Im}_i\left[b^I\right] + j\mathrm{Im}_j\left[b^I\right] + k\mathrm{Im}_k\left[b^I\right],$$

we have:

$$\nabla_{b^I} E = \frac{\partial E}{\partial \mathrm{Re}\left[b^I\right]} + i\frac{\partial E}{\partial \mathrm{Im}_i\left[b^I\right]} + j\frac{\partial E}{\partial \mathrm{Im}_j\left[b^I\right]} + k\frac{\partial E}{\partial \mathrm{Im}_k\left[b^I\right]} \quad (13)$$

$$\nabla_{b^I} E = -\Big\{\,(\mathrm{Re}\,[h] > 0) \times \mathrm{Re}\left[\nabla_{w^{II}} E \cdot w^{II*}\right] + i\,(\mathrm{Im}_i\,[h] > 0)\, \times \mathrm{Im}_i\left[\nabla_{w^{II}} E \cdot w^{II*}\right] + j\,\left(\mathrm{Im}_j\,[h] > 0\right) \times \mathrm{Im}_j\left[\nabla_{w^{II}} E \cdot w^{II*}\right] + k\,(\mathrm{Im}_k\,[h] > 0) \times \mathrm{Im}_k\left[\nabla_{w^{II}} E \cdot w^{II*}\right]\Big\} \quad (14)$$

For the weights,

$$w^I = \mathrm{Re}\left[w^I\right] + i\mathrm{Im}_i\left[w^I\right] + j\mathrm{Im}_j\left[w^I\right] + k\mathrm{Im}_k\left[w^I\right]$$

we have:

$$\nabla_{w^I} E = \frac{\partial E}{\partial \mathrm{Re}\left[w^I\right]} + i\frac{\partial E}{\partial \mathrm{Im}_i\left[w^I\right]} + j\frac{\partial E}{\partial \mathrm{Im}_j\left[w^I\right]} + k\frac{\partial E}{\partial \mathrm{Im}_k\left[w^I\right]} \quad (15)$$

$$\nabla_{w^I} E = -u^* \cdot \nabla_{b^I} E \quad (16)$$

The real-valued RMSProp gradient update equation given by [41]:

$$\theta_{k+1} = \theta_k - \frac{\eta}{\sqrt{v_\theta\,(k)} + \varepsilon} \times \nabla_\theta E \quad (17)$$

where $v_\theta\,(k+1) = \beta_2 v_\theta\,(k) + (1-\beta_2)\,(\nabla_\theta E)^2$, $\eta, \beta_2 \in [0, 1]$, θ can be any weight or bias of the QVNN and ε is a small value that is set to equal 10^{-8}. The moving average parameter β_2 should be strictly greater than 0 and less than 1. RMSprop converges if β_2 is large enough, i.e., near to one. $\beta_2 = 0.999$ was found to be optimal for most applications.

In the proposed forecasting model, we extend Eq. (17) into the quaternion domain and equip it with a softplus activation function. The softplus RMSProp gradient update is computed using [35]

$$\theta_{k+1} = \theta_k - \frac{\eta}{softplus\left(\sqrt{v_\theta\,(k)}\right)} \times \nabla_\theta E \quad (18)$$

In this paper, we use a quaternion softplus function, which has the same form as its real-valued analog and is given by

$$Qsoftplus\left(\sqrt{v_\theta\,(k)}\right) = \frac{1}{\beta} Q\log\left(1 + \exp\left(\beta\sqrt{v_\theta\,(k)}\right)\right) = r_0 + ir_1 + jr_2 + kr_3 \quad (19)$$

where $Q\log$ is the quaternion natural logarithm and is defined as

$$Q\log = \log|q| + \frac{q_p}{|q_p|} ar\cos\frac{x_0}{|q|} \quad (20)$$

where $|q_p| = \sqrt{x_1^2 + x_2^2 + x_3^2}$ is the magnitude of the imaginary component in the quaternion number $q = x_0 + ix_1 + jx_2 + kx_3$, and $|q| = \sqrt{x_0^2 + x_1^2 + x_2^2 + x_3^2}$ is the magnitude of the whole quaternion number. The parameter β(eq. 19) can be any non-negative and non-zero real number.

The square root of a quaternion is computed using Euler rotation angles [42] and is given by

$$(q)^{\frac{1}{2}} = |q|^{\frac{1}{2}} \cdot \left(\left(\cos\frac{\varphi}{2}\cos\frac{\vartheta}{2}\cos\frac{\psi}{2} + \sin\frac{\varphi}{2}\sin\frac{\vartheta}{2}\sin\frac{\psi}{2}\right) + i\left(\sin\frac{\varphi}{2}\cos\frac{\vartheta}{2}\cos\frac{\psi}{2} - \sin\frac{\varphi}{2}\sin\frac{\vartheta}{2}\sin\frac{\psi}{2}\right) + j\left(\cos\frac{\varphi}{2}\sin\frac{\vartheta}{2}\cos\frac{\psi}{2} + \sin\frac{\varphi}{2}\cos\frac{\vartheta}{2}\sin\frac{\psi}{2}\right) + k\left(\cos\frac{\varphi}{2}\cos\frac{\vartheta}{2}\sin\frac{\psi}{2} - \sin\frac{\varphi}{2}\sin\frac{\vartheta}{2}\cos\frac{\psi}{2}\right)\right) \quad (21)$$

where

$$[\varphi,\ \vartheta,\ \psi] = \frac{1}{2} \cdot \begin{bmatrix} \mathrm{atan2}\left(2x_2x_3 + 2x_0x_1,\ x_0^2 - x_1^2 - x_2^2 + x_3^2\right) \\ \text{-asin}\left(2x_1x_3 - 2x_0x_2\right) \\ \mathrm{atan2}\left(2x_1x_2 + 2x_0x_3, x_0^2 + x_1^2 - x_2^2 - x_3^2\right) \end{bmatrix} \quad (22)$$

We use quaternion divisions to divide the quaternion gradient $\nabla_\theta E$ by the term $Qsoftplus\left(\sqrt{v_\theta\,(k)}\right)$. The quaternion division of a quaternion number S is computed using

$1/S = S^*/SS^* = S^*/|S|^2$. The quaternion weights and biases are initialized as random uniformly distributed unit quaternions.

Network parameters η, β_2 and β can be tuned using hyperparameter optimization algorithms such as grid search, random search [44], and Bayesian optimization [45]. In our experiments, we used $\eta = 0.001$, and $\beta_2 = 0.999$, the values recommended in [35].

The performance of the network is strongly influenced by the number of neurons in the hidden layer and the parameter β (eq. 19). The relationship between these two parameters is studied in detail in Section III.

C. PROPOSED FORECASTING METHOD

The proposed forecasting method uses a training stage, where the developed system is trained, and a testing stage where the developed system is used to forecast wind speed values. These stages are summarized below.

1) TRAINING STAGE

1) Obtain a window of n wind speed measurements. This window represents a single training sample. In our experiments, we used a value of $n = 100$.
2) The SWT with two decomposition levels is applied to the wind speed values (Fig. 2).
3) Use the SWT coefficients representing the last sample, i.e., sample n, to train a QVNN to forecast the SWT coefficients representing the next sample $n + 1$.
4) Apply the inverse SWT to a vector composed from the SWT coefficients predicted in step 3 and concatenated with the past n coefficients.
5) Use the $(n + 1)^{\text{th}}$ value of the inverse SWT output as the predicted wind speed value.
6) Compute the mean squared error between the actual and predicted wind speed values.
7) Update the network parameters.
8) Shift the training window by one sample to obtain a new vector of wind speed measurements and iterate until the desired number of epochs is achieved or the mean squared error is below or equal to 0.001.

2) TESTING STAGE

1) Obtain a vector of n past wind speed samples (100 samples in our case).
2) Use the pretrained QVNN to forecast the SWT coefficients representing the next sample $n + 1$ based on the SWT coefficients representing sample n.
3) Apply the inverse SWT to a vector composed of the SWT coefficients predicted in step 2 and concatenated with the past n coefficients.
4) The predicted wind speed value is the $(n + 1)^{\text{th}}$ value of the inverse SWT output vector.

D. EVALUATION CRITERIA

The model's performance is evaluated using the root mean squared error (RMSE), the mean absolute error (MAE), and the mean absolute percentage error (MAPE). These metrics are given as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(y_k - \hat{y}_k\right)^2} \tag{23}$$

$$MAE = \frac{1}{N}\sum_{k=1}^{N}\left|y_k - \hat{y}_k\right| \tag{24}$$

$$MAPE = \frac{100\%}{N}\sum_{k=1}^{N}\left|\frac{y_k - \hat{y}_k}{y_k}\right| \tag{25}$$

where y_k is the k^{th} sample value in y, $\hat{y}_k$ is the kth forecasted value, and N is the total number of samples.

III. RESULTS AND DISCUSSIONS

To evaluate the proposed QVNN-SWT-SRMSProp model, we use four wind datasets, three of which are obtained from the MERRA-2 project for three different areas in China: Peng Lai, Hebei, and Inner Mongolia [46]. The fourth dataset is a real-world wind speed dataset provided by Meteorological Terminal Aviation Routine (METAR) [47].

A. CASE STUDY I: FORECASTING WIND SPEED AT THREE SITES IN CHINA

This example investigates the developed system's performance on data from three different areas in China: Peng Lai, Hebei, and Inner Mongolia. The real-valued neural network (RVNN) or the so-called multilayer perceptron, complex-valued neural network (CVNN), the long-short term memory (LSTM) network, and quaternion-valued neural networks (QVNN) are considered for comparison purposes.

The stopping criteria are either a root mean squared error less than 0.001 or a number of epochs higher than 100. The data are sampled at a standard height of 10 meters, with a temporal resolution of 10 minutes.

Wind speed values sampled at the standard height are first extrapolated to values at the hub height using the following empirical power law [48]:

$$\frac{v}{v_1} = \left(\frac{h}{h_{10}}\right)^{\alpha} \tag{26}$$

where v_1 and v are the wind speeds at the standard height $h_{10} = 10$ meters and the hub height h in meters, respectively, and α is the roughness factor equal to 1/7 [48].

By substituting $h = 50$ meters and the measured wind speed data at the standard height of 10 meters into Eq. (26), the wind speed values at the hub height are obtained. The wind speed data was time-averaged over an hour and time-stamped with the central time of the interval, starting at 00:30 UTC [46]. Wind speed data for the three areas were obtained from May 15, 2019, to May 15, 2020. The first nine months of data (i.e., May 15, 2019, to Feb 15, 2020) are used to train the networks, and the remaining three months of data are used to validate the network's performance.

TABLE 1. Performance of the proposed forecasting method at three Chinese sites: Peng Lai, Hebei, and Inner Mongolia.

Sites	Models	Metrics		
		MAE (m/s)	MAPE (%)	RMSE (m/s)
Peng Lai	RVNN	0.32	15.93	0.39
	CVNN	0.95	32.72	1.02
	QVNN	0.13	3.82	0.17
	LSTM	0.12	4.51	0.18
	RVNN-SWT	0.11	6.33	0.15
	CVNN-SWT	5.12	165.48	5.27
	LSTM-SWT	0.10	3.36	0.18
	QVNN-SWT	0.13	3.83	0.17
	RVNN-SWT-SRMSProp	0.08	4.48	0.11
	CVNN-SWT-SRMSProp	0.24	8.57	0.27
	LSTM-SWT-SRMSProp	0.06	2.07	0.09
	Proposed model (QVNN-SWT-SRMSProp)	**0.03**	**1.25**	**0.06**
Hebei	RVNN	0.63	20.10	0.79
	CVNN	1.31	23.98	1.54
	QVNN	0.17	6.72	0.23
	LSTM	0.29	8.63	0.41
	RVNN-SWT	0.17	6.20	0.22
	CVNN-SWT	3.43	56.21	4.26
	LSTM-SWT	0.26	6.30	0.42
	QVNN-SWT	0.17	6.72	0.25
	RVNN-SWT-SRMSProp	0.15	5.74	0.19
	CVNN-SWT-SRMSProp	0.21	6.26	0.32
	LSTM-SWT -SRMSProp	0.12	3.04	0.19
	Proposed model (QVNN-SWT-SRMSProp)	**0.04**	**1.40**	**0.07**
Inner Mongolia	CVNN	0.91	13.55	1.08
	RVNN	0.59	13.68	0.73
	QVNN	0.73	15.80	0.79
	LSTM	0.46	9.31	0.65
	RVNN-SWT	0.17	4.67	0.22
	CVNN-SWT	10.69	173.09	11.05
	LSTM-SWT	0.18	2.84	0.26
	QVNN-SWT	0.73	15.80	0.79
	RVNN-SWT-SRMSProp	0.11	3.60	0.18
	CVNN-SWT-SRMSProp	0.39	7.89	0.44
	LSTM-SWT -SRMSProp	0.11	2.26	0.17
	Proposed model (QVNN-SWT-SRMSProp)	**0.06**	**1.25**	**0.09**

Wind speed data were scaled to be in the range of 0.1-0.9. The quaternion weights and biases were initialized as random uniformly distributed unit quaternions.

We first studied the effects that the parameter β and the number of hidden neurons have on the performance of the developed model. We trained the developed system on wind speed data from the Inner Magnolia region using numbers of hidden neurons that varied between 5 and 200. In addition, we tested the performance of the forecasting network using different values of the parameter $\beta = \{0.1, 0.5, 1, 5, 50, 100, 500, 1000\}$. The results are shown in Fig. 3, where one can observe a surface of multiple groups of local minima that are close in magnitude at β values between one and five. The overall minimum in terms of MAE (MAE = 0.056) and RMSE (RMSE = 0.90) is achieved using $\beta = 5$ and 110 neurons. The MAPE value at these parameters is close to the MAPE global minimum. This pair of values will be used in the remaining experiments in this paper. In addition, we tested different wavelet families and the Haar wavelet achieved the best results, so we use it in all experiments.

The results from the three sites are presented in Table 1, in which we can see that our model outperforms all other models in all metrics. In this table, methods that don't use the SRMSProp algorithm are optimized using SGD (i.e., a fixed learning rate). We would like to note CVNN-SWT results indicate overfitting, and this can be fixed by reducing the number of hidden neurons n is less than or equal to 60. The performance of the proposed system at this setting remained better than the other forecasting models.

In general, models that use the stationary wavelet transform and the SRMSProp optimizer outperform the remaining methods. Tables 2 and 3 present the results obtained by the proposed model with and without the quaternion softplus RMSProp component. The tables show that the addition of the softplus function reduced overfitting when the number of neuron n = 110 (Table 2), and significantly improved the results when n = 10 (Table 3).

Comparing the proposed model with LSTM-SWT-SRMSProp, which is the model that produced the closest results, the proposed QVNN-SWT- SRMSProp model

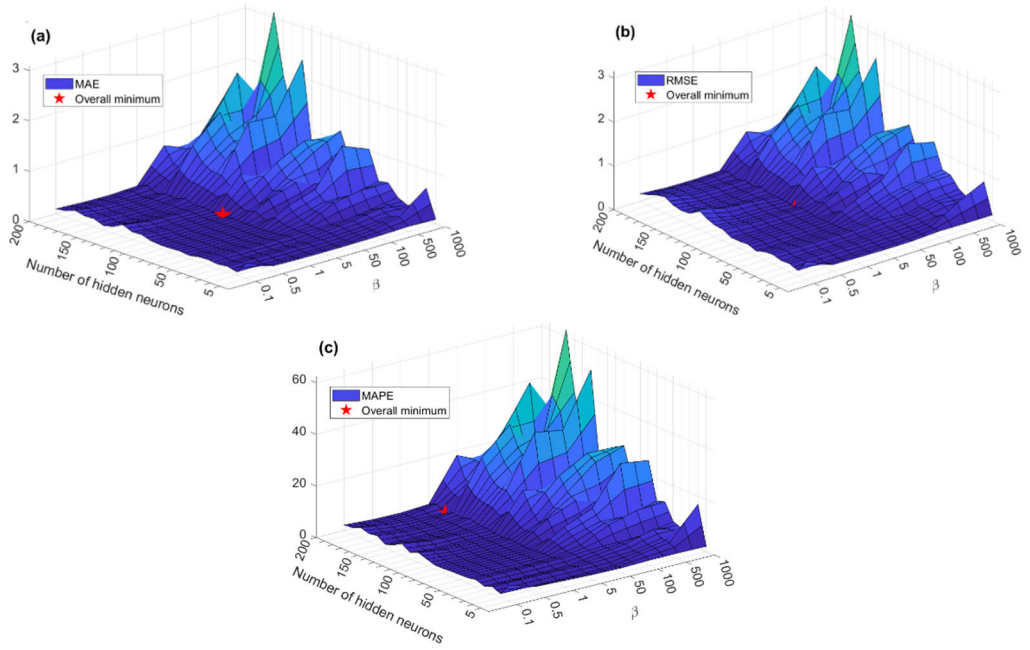

FIGURE 3. The effect of the parameter β and the number of hidden neurons on the performance of the developed model as measured by (a) MAE, (b) MAE (c) MAPE.

TABLE 2. The effect of the softplus quaternion RMSProp optimization method on the performance of the proposed model with 110 hidden neurons.

Sites	Learning algorithm	Metrics		
		MAE (m/s)	MAPE (%)	RMSE (m/s)
Peng Lai	RMSProp	**0.32**	**5.23**	**0.48**
	SRMSProp	**0.03**	**1.25**	**0.06**
Hebei	RMSProp	**0.48**	**12.94**	**0.57**
	SRMSProp	**0.04**	**1.40**	**0.07**
Inner Mongolia	RMSProp	0.63	8.62	0.86
	SRMSProp	**0.06**	**1.25**	**0.09**

TABLE 3. The effect of the softplus quaternion RMSProp optimization method on the performance of the proposed model with 10 hidden neurons.

Sites	Learning algorithm	Metrics		
		MAE (m/s)	MAPE (%)	RMSE (m/s)
Peng Lai	RMSProp	0.07	1.77	0.09
	SRMSProp	**0.03**	**1.04**	**0.06**
Hebei	RMSProp	0.12	2.80	0.19
	SRMSProp	**0.05**	**1.24**	**0.07**
Inner Mongolia	RMSProp	0.13	2.27	0.16
	SRMSProp	**0.07**	**1.40**	**0.11**

improved the MAE, MAPE, and RMSE by (0.06-0.03)/ 0.06 = 50%, (2.07 – 1.25)/2.07 = 40% and (0.09-0.06)/ 0.09 = 33%, respectively, in Peng Lai. In Hebei, the proposed model improves the MAE, MAPE, and RMSE predictions by 66%, 54%, and 63%, respectively, compared with LSTM-SWT-SRMSProp, the closest performing model. In Inner Mongolia, the proposed model improves the MSE, MAPE, and RMSE predictions by 45%, 44%, and 47%, respectively, compared with the LSTM-SWT-SRMSProp model. Fig. 4 shows the validation part of the actual and forecasted wind speed outputs and the corresponding error between them at the Peng Lai site. A zoomed-in view is shown in Fig. 5, in which one can see the performance of the proposed method compared to the SRMSProp optimized RVNN-SWT model.

The proposed model provides better forecasts in regions with smooth transitions, and the error increases in regions with high-frequency content. Similar observations can be observed in the results from Hebei and Inner Mongolia in Fig. 6 and Fig.8.

A zoomed-in comparison between the proposed model and its real-valued counterpart for the Hebei and Inner Mongolia sites is presented in Fig. 7 and 9. The proposed model outperforms the equivalent RVNN-SWT-SRMSPRop model.

Bar graph plots comparing the proposed model with other SRMSProp models in terms of the MAE and RMSE are shown in Fig. 10. The proposed model offers the best wind speed forecasting predictions among the presented systems. This figure indicates that for all sites, the introduction of the SWT improves the correlation values between the measured and forecasted time series and produces outputs with close standard deviation and less root mean square difference.

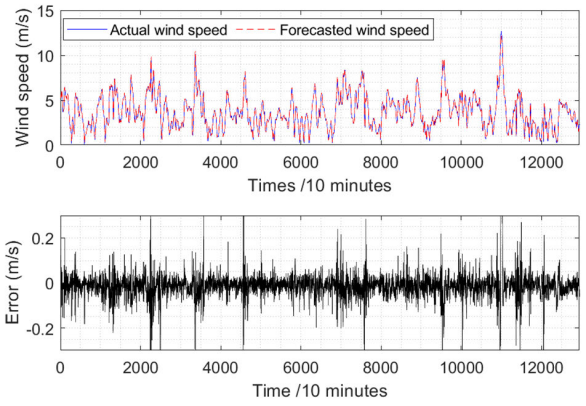

FIGURE 4. Actual and forecasted wind speed using the QVNN-SWT-SRMSProp learning algorithm for the Peng Lai site, and the error between the actual and forecasted wind speed.

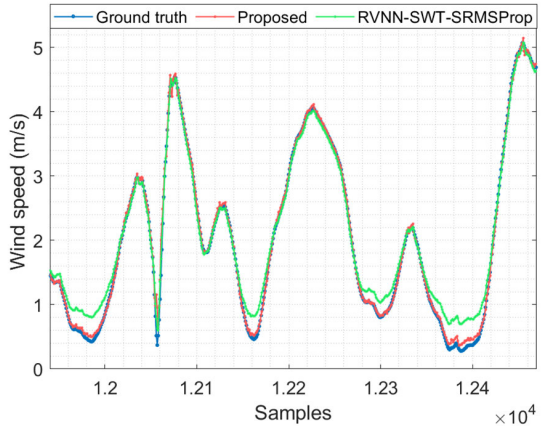

FIGURE 5. A zoomed-in view of the actual and forecasted wind speed using the QVNN-SWT-SRMSProp learning algorithm at the Peng Lai site.

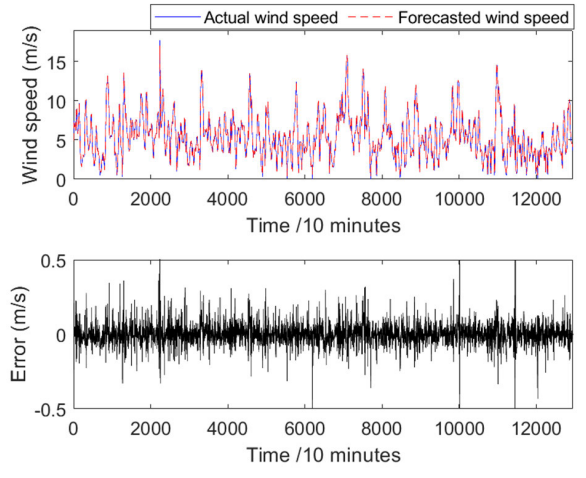

FIGURE 6. Actual and forecasted wind speed using the QVNN-SWT-SRMSProp learning algorithm at the Hebei site, and the error between the actual and forecasted wind speed.

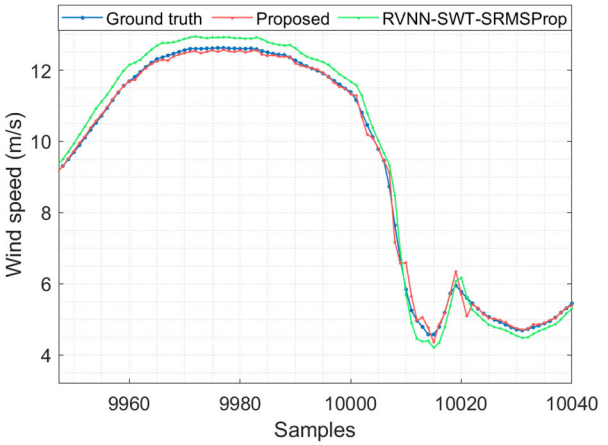

FIGURE 7. A zoomed-in view of the actual and forecasted wind speed using the QVNN-SWT-SRMSProp learning algorithm at the Hebei site.

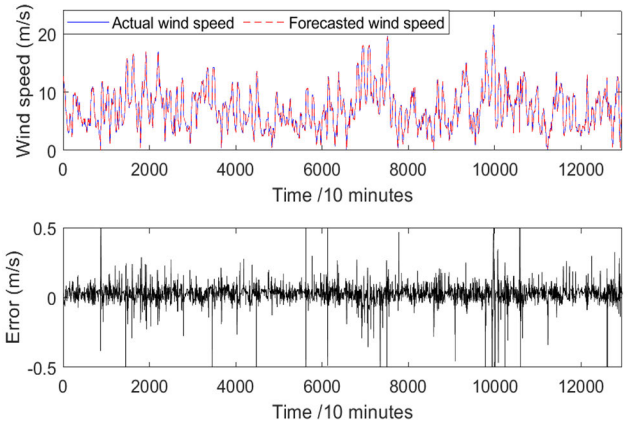

FIGURE 8. Actual and forecasted wind speed using the QVNN-SWT-SRMSProp learning algorithm at the Inner Mongolia site, and the error between the actual and forecasted wind speed.

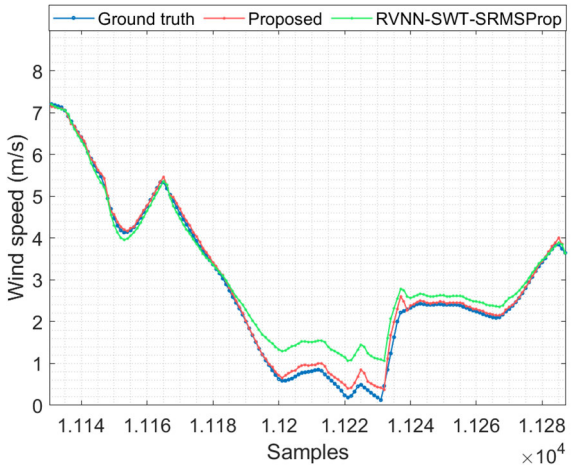

FIGURE 9. A zoomed-in view of the actual and forecasted wind speed using the QVNN-SWT-SRMSProp learning algorithm at the Inner Mongolia site.

B. CASE STUDY II: METEOROLOGICAL TERMINAL AVIATION ROUTINE DATASET

This experiment uses a dataset collected from 57 East Coast stations, including Massachusetts, Connecticut, New York, and New Hampshire weather forecasts provided by the Meteorological Terminal Aviation Routine (METAR) [47]. This dataset consists of 6361 data points that are not incorporated into a standard grid and have an hourly frequency.

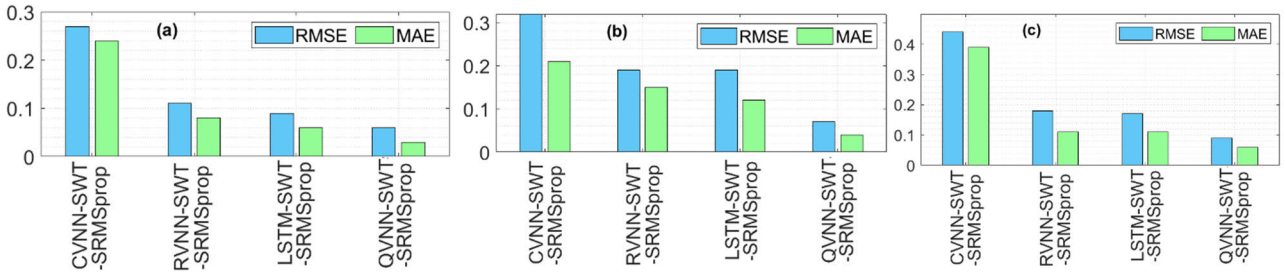

FIGURE 10. Bar graph of the results achieved by SRMSProp methods at the (a) Peng Lai site, (b) Hebei site, and (c) Inner Mongolia site.

TABLE 4. Performance of the proposed forecasting method on the METAR dataset.

Model	RMSE	MAE
Persistence	2.228	1.621
MLP	1.936	1.451
LSTM	1.679	1.270
QVNN	1.628	1.068
CNN [49]	1.509	1.119
DL-STF [47]	1.620	1.180
CoordConv [50]	1.519	1.124
ConvLSTM [54]	1.503	1.110
PSTN [52]	1.716	1.271
PDCNN [50]	1.696	1.265
E2E [51]	1.579	1.179
FC-CNN [51]	1.676	1.251
LI - LW – CNN [47]	1.492	1.106
Proposed	**1.063**	**0.783**

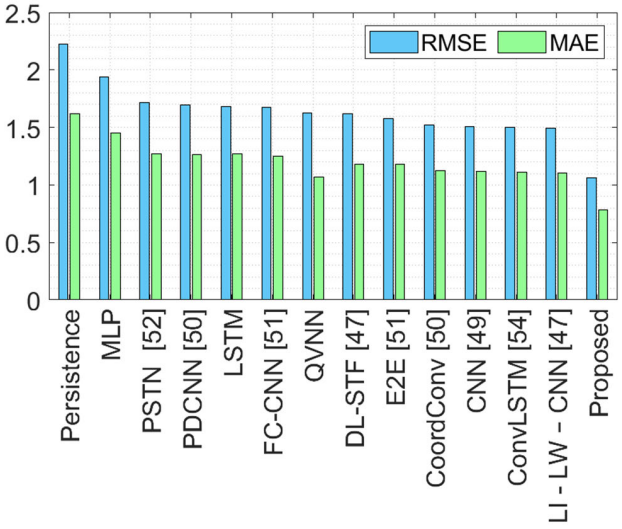

FIGURE 11. Bar graph of the results presented in Table 4.

We use 5700 samples to train the networks, 300 samples for validation, and 361 for testing. This setup allows comparisons with the existing methods in the literature. The data were measured every 6 h, and the goal was to forecast each hour until the next measurement.

We use three widely used prediction models for comparison, namely, the persistence model, multilayer perceptron (MLP), and the long-short term memory (LSTM) network. In addition, we compare the performance of the developed system with other recently published wind speed forecasting models in the literature. These models are: deep learning-based spatiotemporal forecasting (DL-STF) [47], conventional neural network (CNN) [49], learnable inputs-local weights-CNN (LI-LW-CNN) [49], predictive deep convolutional neural network (PDCNN) [50], a combination of a CNN with a fully connected layer (FC-CNN) [50], end-to-end model (E2E) [51], predictive spatiotemporal network (PSTN) [52], CNN with coordinate transform (CoordConv) [53], and convolutional LSTM network (ConvLSTM) [54].

The METAR case results are summarized in Table 4. We computed the Persistence, MLP, and LSTM models; other results are from [47]. We can see that the proposed model outperforms all other forecasting strategies in both evaluation metrics.

The proposed QVNN-SWT-SRMSProp improves the MAE and RMSE by (1.4920 - 1.063) /1.492 $\approx$ 28.8% and (1.106 – 0.783) /1.106 $\approx$ 29.2%, respectively when compared with the LI–LW–CNN, which is the model that produces the nearest results to the proposed model.

A bar graph comparing the proposed model with the remaining models in terms of the MAE and RMSE is shown in Fig. 11.

Despite the effective results achieved by the proposed SWT-QVNN-SRMSProp wind speed forecasting model, we discuss two limitations of the proposed approach.

First, our proposed system is learning-based and may fail when faced with circumstances not observed during training. One potential way to alleviate this issue is to dynamically update the model with new training data in an attempt to increase the size and variability of input data. The second limitation is that our proposed model utilizes quaternion multiplications that can slow the training process.

Computing the Hamilton product of two quaternion neurons requires 28 operations while a single multiplication operation is required to multiply two real neurons [55]. In our experiments, we observe that training a single epoch in a real-valued neural network requires approximately 0.3 seconds and 102.6 seconds in the equivalent quaternion-valued neural network. This problem can be mitigated by developing efficient GPU-based implementations of quaternion multiplications.

In this work, the proposed forecasting model relies only on wind speed data, but other weather parameters like temperature or humidity can be used to improve forecasting performance.

IV. CONCLUSION

In this paper, a novel approach for forecasting wind speed is presented. The developed model uses the stationary wavelet transform and quaternion-valued neural networks. Furthermore, a novel quaternion version of the softplus RMSProp learning algorithm was developed to improve the proposed model's prediction accuracy. Four real-world wind prediction datasets were used to demonstrate the excellent forecasting performance of the proposed model.

Experimental results indicate that the proposed model can effectively forecast wind speeds, particularly over the short term. Therefore, the proposed model is reliable and useful for predicting wind speeds in modern power system management systems.

REFERENCES

[1] H. Liu, C. Yu, H. Wu, Z. Duan, and G. Yan, "A new hybrid ensemble deep reinforcement learning model for wind speed short term forecasting," *Energy*, vol. 202, Jul. 2020, Art. no. 117794.

[2] *Global Wind Statistics*, Council GWE, Global Wind Energy Council, Brussels, Belgium, 2020.

[3] H. Liu and C. Chen, "Data processing strategies in wind energy forecasting models and applications: A comprehensive review," *Appl. Energy*, vol. 249, pp. 392–408, Sep. 2019.

[4] D. R. Drew, D. J. Cannon, J. F. Barlow, P. J. Coker, and T. H. Frame, "The importance of forecasting regional wind power ramping: A case study for the UK," *Renew. Energy*, vol. 114, pp. 1201–1208, Dec. 2017.

[5] M. Liu, Z. Cao, J. Zhang, L. Wang, C. Huang, and X. Luo, "Short-term wind speed forecasting based on the Jaya-SVM model," *Int. J. Electr. Power Energy Syst.*, vol. 121, Oct. 2020, Art. no. 106056.

[6] J. G. Powers, J. B. Klemp, W. C. Skamarock, C. A. Davis, J. Dudhia, D. O. Gill, J. L. Coen, D. J. Gochis, R. Ahmadov, S. E. Peckham, and G. A. Grell, "The weather research and forecasting model: Overview, system efforts, and future directions," *Bull. Amer. Meteorolog. Soc.*, vol. 98, no. 8, pp. 1717–1737, Aug. 2017.

[7] M. Baldauf, A. Seifert, J. Förstner, D. Majewski, M. Raschendorfer, and T. Reinhardt, "Operational convective-scale numerical weather prediction with the COSMO model: Description and sensitivities," *Monthly Weather Rev.*, vol. 139, no. 12, pp. 3887–3905, Dec. 2011.

[8] G. A. Grell, J. Dudhia, D. Stauffer, *A Description of the Fifth-Generation Penn State/NCAR Mesoscale Model (MM5)*, document NCAR/TN-398+STR, University Corporation for Atmospheric Research, 1994, doi: 10.5065/D60Z716B.

[9] K. Zhang, Z. Qu, Y. Dong, H. Lu, W. Leng, J. Wang, and W. Zhang, "Research on a combined model based on linear and nonlinear features—A case study of wind speed forecasting," *Renew. Energy*, vol. 130, pp. 814–830, Jan. 2019.

[10] M. U. Yousuf, I. Al-Bahadly, and E. Avci, "Current perspective on the accuracy of deterministic wind speed and power forecasting," *IEEE Access*, vol. 7, pp. 159547–159564, 2019.

[11] E. Cadenas, W. Rivera, R. Campos-Amezcua, and C. Heard, "Wind speed prediction using a univariate ARIMA model and a multivariate NARX model," *Energies*, vol. 9, no. 2, p. 109, Feb. 2016.

[12] V. Radziukynas and A. Klementavicius, "Short-term wind speed forecasting with ARIMA model," in *Proc. 55th Int. Scientific Conf. Power Electr. Eng. Riga Tech. Univ. (RTUCON)*, Oct. 2014, pp. 145–149.

[13] A. Shamshad, M. A. Bawadi, W. M. A. W. Hussin, T. A. Majid, and S. A. M. Sanusi, "First and second-order Markov chain models for synthetic generation of wind speed time series," *Energy*, vol. 30, no. 5, pp. 693–708, 2005.

[14] F. Cassola and M. Burlando, "Wind speed and wind energy forecast through Kalman filtering of numerical weather prediction model output," *Appl. Energy*, vol. 99, pp. 154–166, Nov. 2012.

[15] H. Hu, L. Wang, and R. Tao, "Wind speed forecasting based on variational mode decomposition and improved echo state network," *Renew. Energy*, vol. 164, pp. 729–751, Feb. 2021.

[16] M. Zhou, B. Wang, S. Guo, and J. Watada, "Multi-objective prediction intervals for wind power forecast based on deep neural networks," *Inf. Sci.*, vol. 550, pp. 207–220, Mar. 2021.

[17] A. Kisvari, Z. Lin, and X. Liu, "Wind power forecasting—A data-driven method along with gated recurrent neural network," *Renew. Energy*, vol. 163, pp. 1895–1909, Jan. 2021.

[18] R. Cai, S. Xie, B. Wang, R. Yang, D. Xu, and Y. He, "Wind speed forecasting based on extreme gradient boosting," *IEEE Access*, vol. 8, pp. 175063–175069, 2020.

[19] R. Wang, J. Wang, and Y. Xu, "A novel combined model based on hybrid optimization algorithm for electrical load forecasting," *Appl. Soft Comput.*, vol. 82, Sep. 2019, Art. no. 105548.

[20] X. Liu, H. Zhang, X. Kong, and K. Y. Lee, "Wind speed forecasting using deep neural network with feature selection," *Neurocomputing*, vol. 397, pp. 393–403, Jul. 2020.

[21] 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, Apr. 2020, Art. no. 100601.

[22] A. Khosravi, R. N. N. Koury, L. Machado, and J. J. G. Pabon, "Prediction of wind speed and wind direction using artificial neural network, support vector regression and adaptive neuro-fuzzy inference system," *Sustain. Energy Technol. Assessments*, vol. 25, pp. 146–160, Feb. 2018.

[23] H. B. Azad, S. Mekhilef, and V. G. Ganapathy, "Long-term wind speed forecasting and general pattern recognition using neural networks," *IEEE Trans. Sustain. Energy*, vol. 5, no. 2, pp. 546–553, Apr. 2014.

[24] R. Rahmani, R. Yusof, M. Seyedmahmoudian, and S. Mekhilef, "Hybrid technique of ant colony and particle swarm optimization for short term wind energy forecasting," *J. Wind Eng. Ind. Aerodyn.*, vol. 123, pp. 163–170, Dec. 2013.

[25] Y. Zhang, S. Gao, J. Han, and M. Ban, "Wind speed prediction research considering wind speed ramp and residual distribution," *IEEE Access*, vol. 7, pp. 131873–131887, 2019.

[26] Z. Sun and M. Zhao, "Short-term wind power forecasting based on VMD decomposition, ConvLSTM networks and error analysis," *IEEE Access*, vol. 8, pp. 134422–134434, 2020.

[27] H. Liu, H.-Q. Tian, D.-F. Pan, and Y.-F. Li, "Forecasting models for wind speed using wavelet, wavelet packet, time series and artificial neural networks," *Appl. Energy*, vol. 107, pp. 191–208, Jul. 2013.

[28] Z. Peng, S. Peng, L. Fu, B. Lu, J. Tang, K. Wang, and W. Li, "A novel deep learning ensemble model with data denoising for short-term wind speed forecasting," *Energy Convers. Manage.*, vol. 207, Mar. 2020, Art. no. 112524.

[29] H. Liu, X.-W. Mi, and Y.-F. Li, "Wind speed forecasting method based on deep learning strategy using empirical wavelet transform, long short term memory neural network and Elman neural network," *Energy Convers. Manage.*, vol. 156, pp. 498–514, Jan. 2018.

[30] S. N. Singh and A. Mohapatra, "Repeated wavelet transform based ARIMA model for very short-term wind speed forecasting," *Renew. Energy*, vol. 136, pp. 758–768, Jun. 2019.

[31] S. Pei, H. Qin, Z. Zhang, L. Yao, Y. Wang, C. Wang, Y. Liu, Z. Jiang, J. Zhou, and T. Yi, "Wind speed prediction method based on empirical wavelet transform and new cell update long short-term memory network," *Energy Convers. Manage.*, vol. 196, pp. 779–792, Sep. 2019.

[32] M. Stephane, *A Wavelet Tour of Signal Processing*. Amsterdam, The Netherlands: Elsevier, 1999.

[33] T. Parcollet, M. Morchid, and G. Linarès, "A survey of quaternion neural networks," *Artif. Intell. Rev.*, vol. 53, no. 4, pp. 2957–2982, Apr. 2020.

[34] L. S. Saoud, R. Ghorbani, and F. Rahmoune, "Cognitive quaternion valued neural network and some applications," *Neurocomputing*, vol. 221, pp. 85–93, Jan. 2017.

[35] Q. Tong, G. Liang, and J. Bi, "Calibrating the adaptive learning rate to improve convergence of Adam," 2019, *arXiv:1908.00700*. [Online]. Available: http://arxiv.org/abs/1908.00700

[36] G. P. Nason and B. W. Silverman, "The stationary wavelet transform and some statistical applications," in *Wavelets and Statistics* (Lecture Notes in Statistics), vol. 103, A. Antoniadis and G. Oppenheim, Eds. New York, NY, USA: Springer, 1995, pp. 281–299, doi: 10.1007/978-1-4612-2544-7_17.

[37] S. Supratid, T. Aribarg, and S. Supharatid, "An integration of stationary wavelet transform and nonlinear autoregressive neural network with exogenous input for baseline and future forecasting of reservoir inflow," *Water Resour. Manage.*, vol. 31, no. 12, pp. 4023–4043, Sep. 2017.
[38] L. Saad Saoud, F. Rahmoune, V. Tourtchine, and K. Baddari, "A novel method to forecast 24 h of global solar irradiation," *Energy Syst.*, vol. 9, no. 1, pp. 171–193, Feb. 2018.
[39] T. Nitta, "A quaternary version of the back-propagation algorithm," in *Proc. Int. Conf. Neural Netw. (ICNN)*, Nov. 1995, pp. 2753–2756.
[40] L. Saad Saoud, F. Rahmoune, V. Tourtchine, and K. Baddari, "Fully complex valued wavelet network for forecasting the global solar irradiation," *Neural Process. Lett.*, vol. 45, no. 2, pp. 475–505, Apr. 2017.
[41] G. Hinton, "Neural networks for machine learning—Lecture 6a—Overview of mini-batch gradient descent," Univ. Toronto, Toronto, ON, Canada, 2012. Accessed: Dec. 20, 2020. [Online]. Available: https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf
[42] J. Diebel, "Representing attitude: Euler angles, unit quaternions, and rotation vectors," *Matrix*, vol. 58, nos. 15–16, pp. 1–35, 2006.
[43] X. Zhu, Y. Xu, H. Xu, and C. Chen, "Quaternion convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 631–647.
[44] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 281–305, 2012.
[45] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, vol. 2, Dec. 2012, pp. 2951–2959.
[46] *MERRA-2 tavg1_2d_slv_Nx: 2d,1-Hourly, Time-Averaged, Single-Level, Assimilation, Single-Level Diagnostics V5.12.4*, Global Modeling and Assimilation Office (GMAO), Goddard Earth Sciences Data and Information Services Center (GES DISC), Greenbelt, MD, USA, 2015, Accessed: Jul. 20, 2020.
[47] A. Ghaderi, B. M. Sanandaji, and F. Ghaderi, "Deep forecast: Deep learning-based spatio-temporal forecasting," 2017, *arXiv:1707.08110*. [Online]. Available: http://arxiv.org/abs/1707.08110
[48] R. C. Bansal, T. S. Bhatti, and D. P. Kothari, "On some of the design aspects of wind energy conversion systems," *Energy Convers. Manage.*, vol. 43, no. 16, pp. 2175–2187, Nov. 2002.
[49] A. Uselis, M. Lukoševičius, and L. Stasytis, "Localized convolutional neural networks for geospatial wind forecasting," *Energies*, vol. 13, no. 13, p. 3440, Jul. 2020.
[50] Q. Zhu, J. Chen, L. Zhu, X. Duan, and Y. Liu, "Wind speed prediction with spatio–temporal correlation: A deep learning approach," *Energies*, vol. 11, no. 4, p. 705, Mar. 2018, doi: 10.3390/en11040705.
[51] R. Yu, Z. Liu, X. Li, W. Lu, D. Ma, M. Yu, J. Wang, and B. Li, "Scene learning: Deep convolutional networks for wind power prediction by embedding turbines into grid space," *Appl. Energy*, vol. 238, pp. 249–257, Mar. 2019.
[52] Q. Zhu, J. Chen, D. Shi, L. Zhu, X. Bai, X. Duan, and Y. Liu, "Learning temporal and spatial correlations jointly: A unified framework for wind speed prediction," *IEEE Trans. Sustain. Energy*, vol. 11, no. 1, pp. 509–523, Jan. 2020, doi: 10.1109/TSTE.2019.2897136.
[53] R. Liu, J. Lehman, P. Molino, F. Petroski Such, E. Frank, A. Sergeev, and J. Yosinski, "An intriguing failing of convolutional neural networks and the CoordConv solution," 2018, *arXiv:1807.03247*. [Online]. Available: http://arxiv.org/abs/1807.03247
[54] X. Shi, Z. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS) 28th Int. Conf. Neural Inf. Process. Syst.*, vol. 1, Dec. 2015 pp. 802–810.
[55] T. Parcollet, M. Ravanelli, M. Morchid, G. Linarès, C. Trabelsi, R. De Mori, and Y. Bengio, "Bengio quaternion recurrent neural networks," in *Proc. ICLR*, Nouvelle Orleans, LA, USA, May 2019, pp. 1–19.

LYES SAAD SAOUD received the D.Sc. and M.C.A. degrees (empowerment to direct research) in physics, specialty renewables energies from the University of Boumerdès, Algeria, in January 2015 and February 2017, respectively. He was an Associate Professor with the University of Boumerdès. From February 2019 and October 2019, he became the Head and the Scientific Member of the Department of Electrical Systems Engineering and a Scientific Member of the Faculty of Technology, Boumerdès. He is currently an Engineer and Magister in electronics. His recent research interests include advanced neural networks, fuzzy logic, hybrid systems, renewable energy, sustainability and the environment, energy engineering, and cybersecurity. He has served as a member for the Technical Committee for several international specialized conferences. He is a Reviewer of IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE ACCESS, *Transactions of the Institute of Measurement and Control* (SAGE Publications), *Journal of Renewable and Sustainable Energy* (the American Institute of Physics), *Neurocomputing* (Elsevier), and *Neural Processing Letters* (Springer).

HASAN AL-MARZOUQI (Senior Member, IEEE) received the bachelor's (Hons.) and M.S. degrees in electrical and computer engineering from Vanderbilt University, Nashville, TN, USA, in 2004 and 2006, respectively, and the Ph.D. degree in electrical and computer engineering from Georgia Institute of Technology, in 2014. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Khalifa University of Science and Technology, Abu Dhabi, United Arab Emirates. His research interests include machine learning, image and video processing, and digital rock physics. He served as an Organizing Committee Member and the Challenge Sessions Co-Chair for ICIP 2020 in Abu Dhabi and a Technical Program Committee Member in EUVIP 2020 in Paris. He is an active reviewer in many international conferences and journals.

MOHAMED DERICHE (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from the National Polytechnic School, Algeria, and the M.Sc. and Ph.D. degrees in signal processing from the University of Minnesota, in 1994. He worked at Queensland University of Technology, Australia, before joining King Fahd University of Petroleum and Minerals (KFUPM) in Dhahran, Saudi Arabia, in 2001, where he led the Signal Processing Group. In September 2021, he joined Ajman University, United Arab Emirates. He has published more than 300 papers in multimedia signal and image processing. He has supervised more than 40 M.Sc. and Ph.D. students. His current research interests include different aspects of multimedia signal and image processing, seismic applications, biomedical signal processing, and diverse applications of artificial intelligence and machine learning. He was a recipient of the IEEE Third Millennium Medal. He also received Shauman Best Researcher Award, and both the Excellence in Research and Excellence in Teaching Awards at KFUPM. He delivered numerous invited talks and chaired several conferences, including GlobalSIP-MPSP, IEEE Gulf (GCC), Image Processing Tools and Applications, and TENCON (a Region ten conference). He is the Founding Associate Editor for the *International Journal of Sensors, Transducers and Instrumentation Systems*. He serves as an Associate Editor for *EURASIP Journal for Image and Video Processing*, and was with *Electronics Letters* (IET), *International Journal of Digital Signals and Smart Systems*, *International Journal on Imaging and Graphics*, and *Electronic Letters on Computer Vision and Image Analysis*.