IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 8 September 2022, accepted 28 September 2022, date of publication 3 October 2022, date of current version 6 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3211312

RESEARCH ARTICLE

Time Series Forecasting With Volatility Activation Function

FURKAN KAYIM AND ATINÇ YILMAZ
Computer Engineering Department, Faculty of Engineering and Architecture, Beykent University, 34398 Istanbul, Türkiye
Corresponding author: Furkan Kayim (kayimfurkan@gmail.com)

ABSTRACT Time series forecasting is the method of predicting future values of a model by reviewing its past data. Various models like traditional approaches, statistical methods, moving average, ARIMA, RNN's, or XGBoost may also be applied. Activation functions are the primary most important hyper parameters or artificial neural networks that decide whether a neuron will be active or not. However, the most widely used sigmoid and ReLU activation functions include the problems of linearity, gradient protection, vanishing gradient and data convergence. Solution of these problems is significant for the development of activation functions which are one of the most important hyper parameters for artificial neural networks. A new activation function is suggested to generate a solution for the problems existing in the activation functions within the scope of this study. A hybrid model has been created from the RNN and LSTM algorithms and applied on different time series data in order to implement this suggested activation function called as the volatility activation function. The volatility activation function has been compared with the literature studies, the conditions for being a function have been proved and its applicability has been demonstrated by means of financial instrument and electrical transformer temperature data. As a result of the study, the characteristics of the volatility activation function have been presented; additionally its applicability, validity and proof has been performed. Furthermore, it has been proved that the problems found in the activation functions used in the literature are not existing in the volatility activation function. It has been verified that the volatility activation function is functional on time series. To demonstrate the feasibility of the volatility activation function, it has been applied to three different time series problems. As a result of the study conducted on the electrical transformer temperature estimation, MSE = 0.362 and MAE = 0.448 were obtained; namely these results are similar to the literature studies. In the test results with ounce gold data, the accuracy rate increased by 0,1. In the test results with Australian rain forecast data, the accuracy rate increased by 0,2. As a result, a new activation function is suggested for the deep learning.

INDEX TERMS Activation functions, artificial intelligence, deep learning, time series forecasting.

I. INTRODUCTION

Time series forecasting will be conducted by using deep learning within the scope of this study. Time series are the data sets in which data is ordered within a period. On the other hand, time series analysis can be described as to predicting how the series will continue in the future by reviewing the data within this period. A new activation function will be suggested while forecasting the time series and the forecasts will be made with this suggested activation function within the scope of this study. The volatility activation function will be suggested as an alternative solution for the problems of data convergence, linearity, gradient protection, vanishing gradient which are widely existing in the activation functions and obtaining higher forecast results in the time series forecasting. There are data convergence, linearity, gradient protection and vanishing gradient problems in the activation functions.

Activation functions do not work based on specific data. Rather, they work for all data. However, biological neurons are divided into three groups: Afferent neurons, motor neurons and interneurons.

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

- The task of specialized cells known as afferent neurons: they transmit information from outside to the brain through hearing, touch, taste and smell.
- The function of motor neurons is to provide movement by looking at muscle contraction. Interneurons which are completely placed in the nervous system create a circuit adjacent to their association neurons (circuit is fulfilled by local interneurons).
- Transplant interneurons connect the circuit created by a local interneuron existing in a certain region of the brain to a circuit in another region.

The neuron circuits in the brain perform the functions necessary for learning and perception by using these connections. The fact that these features, which are existing in the real neural networks, are not existing in artificial neural networks, is called as the data convergence problem. This problem gives rise to the possibility of inability to analyze the system work and failure of the system in learning.

Linear activation functions are not considered appropriate for complex valuable systems.

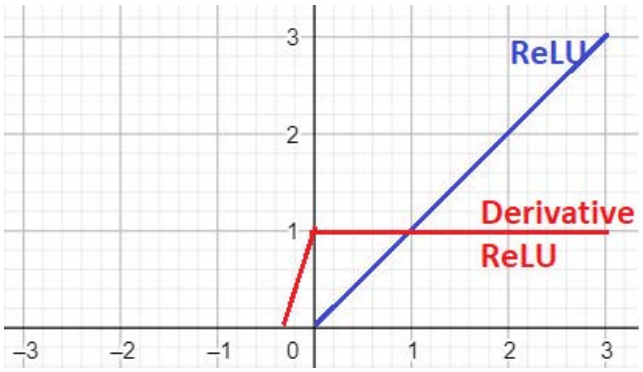

FIGURE 1. ReLU and its derivative.

The problem of linearity is existing on the ReLu activation function given in Figure 1. In order not to cause models working with deep learning to work like linear regression, nonlinear activation functions are preferred in deep neural network architectures [1]. There are many studies on this subject in the literature. The use of linear activation function is a problem because it will lead its model to behave linear. And this means that deep neural networks can no longer handle the complex, non-linear data they used to improve performance.

It has been found by the studies in the literature that the gradient protection feature of ReLU has a derivative of 1 when $x > 0$, that it will no longer be a distinctive advantage in modern architectures. ReLU activation function has a derivative on Figure 1. The derivative of ReLU generates the value 1 for each situation. This situation sets forth a considerable disadvantage and problem.

The error is calculated backward for the training of neural network. We can calculate the derivative for this function. Indeed, we can calculate its training, in other words to what extent it changes in a certain place. If we do it for our existing point on the error curve, we can forecast where to go in order to improve that particular weight. Backpropagation allows us to descend the gradient for all weights. By chaining the existing gradients, we can calculate the gradient for any weight - and consequently calculate the improvements based on backward errors towards the highest flow layer in the network.

The results are found within the range of $[-1,1]$ in the activation functions like Sigmoid, tanh. Convergence to zero takes place as a result of using the backpropagation algorithm. Gradient descent process is used to train Artificial Neural Networks. The reason why the gradient descent consists of a chain-like backward propagation step is to reduce the loss in each period and to obtain the change in weights. Considering a two layered network, this first layer is represented as $f1(x)$ and the second layer is represented as $f2(x)$. The general network is as following: $o(x) = f2(f1(x))$. If we calculate the weights in the course of back transition, we would obtain $o'(x) = f2(x)^*f1'(x)$ where $f1(x)$ is a compound function composed of Act $(W1^*x1 + b1)$, where the Act is the activation function following the layer 1. If we apply the chain rule again, we can clearly see that $f1'(x) = \text{Act}.(W1^*x1 + b1)^*x1$, that means it directly depends on the activation value. Now imagine that such a chain rule passes through multiple layers in the course of backpropagation. After checking that the Act() value is between 0 and 1, the first layer gradient is calculated as a result of multiplying a few such values.

As a result, the gradient of the first layers decreases and becomes unlearned. In other words, the activation value is shifted to zero and the gradient tends to disappear. The disappearing gradient problem causes the termination of the derivative process, that is, the end of the learning process [2], [3]. It is therefore preferable that the activation functions do not shift their gradient to zero.

In this study, a new activation function that will provide a solution for all these problems existing in the activation functions is introduced. With the proposed new activation function, a powerful alternative activation function is presented. In the second part, related studies are explained and the differences of the proposed method from the literature are presented. In the third section, the proposed method is explained. In the fourth chapter, the application of the proposed method on three different time series problems is presented. In the fifth section, the results are presented.

II. RELATED WORK

Marques et al. [4] suggests ePL-KRLS-FSM as a new class of fuzzy modelling approach developed for time series forecasting in his study. With this method, he suggested a model that could predict chaotic data more precisely, allowing inaccuracies in the data to be better addressed. Comparisons with different time series, additionally some comparisons with traditional forecasting models have been conducted in the model. In consequence, it has been evidenced that the suggested model is competitive and more consistent than the models being compared.

Fanjiang et al. [5] conducted his research on the quality of service (QoS) time series forecasting for Web services (WS'ler) and suggested a genetic programming based approach in order to solve the multi-step QoS time series forecasting problem. Based on the QoS time series data set, the suggested approach has been verified and compared with several traditional methods in order to evidence its superiority in terms of accuracy.

Bash et al. [6], in his study, suggests that India is located in the tropical wet and dry zone and has revealed a study that forecasts annual rainfall during the monsoon season. He used several time series forecast models in order to predict the amount of rainfall in his study like Seasonal Autoregressive Integrated Moving Average (SARIMA), Holt's Winter Seasonal Method (HWSM), Generalized Auto-Regressive Conditional Heteroskedasticity (GARCH) and Holt's Linear Trend Method (HLTM). He evaluated the models with the methods of RMSE (Root Mean Square Error), Mean Absolute Error (MAE) and Mean Squared Error (MSE) as the evaluation parameters used to perform a comparative analysis of different Time Series forecasting models and he found that HWSM model is the most successful method when compared with other models.

Saad [7] estimated the number of COVID-19 cases caused by the impact of the coronavirus in his study. In conducting his study of data from India, Canada, the United Kingdom, Australia, Egypt, and the United States, he used the Seasonal Auto-Regressive Integrated Moving Average (SARIMA) method as opposed to an optimized version of the cross-sectional multivariate epidemiological model called SEIRD. Non-Linear L-BFGS-B Installed as SEIRD. As a result of his studies, he achieved more successful estimation results with SEIRD.

Liu and Zhang proposed a time series estimation method called Convolution Nuclear Norm Minimization (CNNM) by examining the perspective of compressed perception from time series data. Guangcan [8], on the other hand, studied low-order signals among regular signals. He integrated an orthonormal transformation into CNNM while doing his work. As a result, he tried to approach the problems and suggested the Learning Based CNNM (LbCNNM) method. He found that LbCNNM shows superior performance as a result of extensive experiments held on 100,452 real-world time series from the Time Series Data Library (TSDL) and the M4 Competition (M4).

Gupta [9] put forward a study on the forecasting solar power generation for future planning of smart grid integration due to the ever-increasing number of solar power plants and the decrease in dependence on fossil fuels in the energy sector. While using the Machine Learning algorithms such as Facebook (FB) Prophet and Extreme Gradient Boost (XGB) to forecast solar power generation on a monthly and weekly basis, he concluded in his study that the XGB model is more effective than the FB model in terms of better forecasting and better compliance. He calculated the RMSE, MAPE and MAE parameters in order to check the performance of RMSE, MAPE and MAE model.

Malhan [10] has compared three decomposition based machine learning algorithms in order to evaluate the performance of wind energy in multi-step univariate time series forecasting and the models he selected were STL-ARIMA, CEEMD-BiLSTM and CEEMDAN-BiLSTM. The accuracy of these models has been tested for different time frames ranging between short term namely one day before and long term namely three years ago. The results evidenced that CEEMDAN-BiLSTM is the most appropriate model for short term wind energy generation forecast. Another result he found is that STL-ARIMA gives better wind energy forecasts in the long term.

Maya et al. [11] has set forth a study that successfully solves the common problems of time series forecasts based on neural networks by combining meta-learning and transfer-learning methods to solve the problem of lost data in time series forecasting. He forecasted Mexico City's air pollution with incomplete data by analyzing the strong and weak convergences of his suggested algorithms. His work has evidenced that meta-transfer learning has a very good forecasting accuracy.

Demidova et al. [12], in her study, discussed a biology-inspired approach to Long short term memory (LSTM) loss function optimization and compared the performance of different LSTM networks trained with backpropagation and using biology-inspired algorithms including Genetic Algorithm, Particle Swarm optimization, and Fish School Search (FSS). In the end, she evidenced that the exponential step distortion (ETFSS) and FSS algorithms have given superior results compared to GA and PSO in most of the optimization problems.

Sarıkaa et al. [13] used time series forecasting models in her study because the cloud sources forecasting is time dependent and placed different time series models and community techniques in order to forecast resource use for successive hours. Different time series analysis techniques have been applied in order to examine data set and select the appropriate model. As a result of her study, she forecasted several source types like CPU use, disk reading periods, disk writing periods and memory use.

Mouine et al. [14] has analyzed time series forecasting models by using ARIMA, Prophet and LSTM in order to forecast the number of virtual machines which need cloud source monitoring data in her study conducted on description of a forecast model that is effective on game session workloads for which a certain time period is given. As a result of her study, she found LSTM is the most accurate model in forecasting the demand of virtual machines in terms of RMSE and MAE.

Carpenter et al. [15] has performed time series forecasting by using COVID-19 pandemics data in his study. While doing this, he suggested a model which can make global scale forecasts only after the first four weeks of the pandemics. He utilized LSTM, GRU algorithms in his study. He compared

how LSTM and GRU performs by using different regulators. In consequence, he evidenced it reduces RMSE by 3% in datasets with only 28 days of training.

Lin [16] made a new proposal for time series estimation. His proposal is called SSDNet, which is a new method of deep learning. SSDNet looks at previous steps of seasonality and trend data when making time series forecasting. It presents a different approach by combining space models and transformer architecture. SSDNet has performed a performance evaluation. In doing so, he used five data sets. It looked at accuracy and speed in the test results. It has proven that SSDNet outperforms deep learning and statistical methods and can provide significant seasonality and trend components with study.

Zúñiga et al. [17] has suggested a new time series forecasting method called ForGAN in his study and wanted to indicate that a probability forecast of the time series called as hostile producer network is successfully used for one step forward forecasts. A modified ForGAN architecture having multiple outputs has been suggested in order to perform multi-step forecasts with his study.

Kellerman et al. [18] analyzed large amounts of sensor data coming from machines in the time domain in his study. He introduced a preprocessing method he called as ''temporal resolution warping'' (TRW) and to validate his methods, he used feedforward and convolutional neural networks including residual layers to forecast the reference time series of different applications. He has shown that training accelerates by more than 26% with his techniques.

In Arporn's [19] study, empirical wavelet transform (EWT) was applied to decompose the original series into several subseries containing unique features at different frequency horizons. Then, ARIMA used ANN and moving average filter (MA) and subsequences to perform forecasting tasks. To do this, he used four publicly available real-world datasets and compared them to six other benchmarking models. As a result of his studies, he proved that the proposed model provides a high prediction accuracy.

Jacoby [20] proposes a new time series forecasting model called Adaptive Fuzzy Forecasting (AFP) in Commercial Microwave interconnects (CMLs). The time-series forecasting models he proposes are based on the assumption that he regularly uses a model with variations of Autoregressive Integrated Moving Average (ARIMA) and Recurrent Neural Network (RNN) in which the entire data set follows the same Data Generation Process (DGP). The results showed that the proposed methods increase the performance of the prediction model by 30-40%, depending on the data availability and the specific model.

Samal et al. [21] used air pollution forecasting in his study. While doing this, he has utilized deep learning and aimed to forecast multiple variables at the same time. He performed air pollution forecasting in Gucheng location as one of Beijing's polluted cities, made an air pollution forecast and became effective in air pollution forecasting according to results of his study.

Choi et al. [22] performed text-independent speaker verification in his study. He used VoxCeleb and Speakers in the Wild datasets as the dataset in his study. He evidenced that his suggested method improves the text-independent speaker verification performance under the short-statement condition.

Liao et al.[23] made temperature time series forecast with deep learning in his study. In his research, he generated probability forecasts by displaying the original weather data as numerical time series. He used the LSTM algorithm in his research and compared his study with other models.

In general, while statistical methods are applied for the solution of time series problems in the studies existing in the literature, the methods in which artificial intelligence techniques are applied also exist. A new approach is not set forth in most of the studies, however some research on comparison of the methods existing in the literature have been conducted. The contribution in artificial neural networks is very little in the studies by which a new approach is found. In scope of this study, time series forecasting has been performed with the artificial intelligence techniques. Unlike the studies existing in the literature, a new activation function has been applied while performing time series forecasting. The suggested method has been tested with different data sets and its applicability, accuracy and proof has been put forth with different approaches.

III. MATERIALS AND METHODS

In scope of the study, sigmoid, ReLU and volatility activation functions have been applied on the hybrid model composed of LSTM and RNN algorithms and the activation function results have been compared.

Sigmoid activation function is stated as (1) and has a S-shape graph [21].

$$y = \frac{1}{e^{-x}} \tag{1}$$

The ReLU activation function is as shown in (2).

$$y = \begin{cases} 0, & X < 0 \\ X, & X \geq 0. \end{cases} \tag{2}$$

LSTM is a RNN type that has the same kind of input and output. However, RNN is different from LSTM. This difference is that LSTM is an input gate, a forget gate and an output gate. In this study, Recurrent Neural Network (RNN) and Long Short-term Memory (LSTM) algorithms are used in hybrid form to make time series estimation.

A. PROPOSED VOLATILITY ACTIVATION FUNCTION

The volatility function used in the suggested method may be applied for every data that involves a period and value. It is possible to apply this function since there is a period and value in time series. Its applicability and validity have been demonstrated on electrical transformer data and financial instruments. The expression has been strengthened with simple examples.

Volatility means statistical measure of allocation for a certain period for time series. Volatility is calculated by using the standard deviation or variance of time series data. Volatility is used for convergence of activation functions to time series in scope of the study. Periodical volatility formula can be calculated by finding square root of the periodical time series data variance. A time value like day, week, month, year, hour, minute can be used for the period given in the formula.

The volatility activation function formula is as shown in equation (3) and equation (4).

$$\text{Volatilite} = \sqrt{\frac{\sum (P_{mean} - P_i)^2}{n}} \tag{3}$$

Volatility formula consist of P_{mean}, P_i and n is constant. Let's consider $P_i = x_i$.

P_{mean} depends on x_i .. Therefore $P_{mean} = \bar{x}$.

$$f(x_1, \ldots, x_n) = \sqrt{\frac{\sum (\bar{x} - x_i)^2}{n}} \tag{4}$$

The variables used in the Volatility activation function described in Equation (3) are as follows:

Period = Time interval between start and end times.

$y = P_i$ = Time series data at i time

$x = P_{mean}$ = Mean value of the time series data within the period

P_{total} = Total value of the time series data within the period.

n = Number of time series existing in the period.

$P_{squaresum}$ = Square sum value of the time series data within the period.

Variance = Time series variance.

Volatility = Volatility value for the period.

Start and end times and time series data expected from outside.

The volatility formula of a certain time series data can be obtained by using the following steps:

1. Firstly P_{total} is calculated, time series data within the period is summed.

2. Then, P_{mean} is calculated by determining the mean of time series data. $P_{mean} = P_{total}/n$.

3. Then, the difference between each period value and mean value, $P_i - P_{mean}$ is calculated.

4. Then, the square of all deviations is calculated, $(P_{mean} - Pi)^2$.

5. Then, $P_{squaresum}$ namely the sum of all square deviations is found, $\sum (P_{mean} - Pi)^2$.

6. Variance = $P_{squaresum}/n = \sum (P_{mean} - P_i)^2 / n$

7. Then, period volatility or standard deviation is calculated by finding square root of time series variance.

8. Volatility = $\sqrt{}$ Variance = $\sqrt{(\sum (P_{mean} - P_i)^2 / n)}$

Extended volatility = $\sqrt{}$ Different Periods $^*\sqrt{(\sum (P_{mean} - P_i)2 / n)}$

In the time series X = {X1,X2,…,Xn} consists of the xi of the elements in the time series. P_{mean} is their average.

The average value obtained is squared by adding the differences from the values in the individual time series. This value obtained is divided by the number of elements in the time series (n).

In the previous paragraph, how the P_{mean} is calculated and how the volatility is calculated is calculated step by step. In addition, step-by-step volatility calculation using real data is explained under the heading of sample of volatility formula.

B. SAMPLE OF VOLATILITY FORMULA

The volatility activation function was calculated on the time series in Table 1. There is 10-day data of EUR/USD on Table 1 and these data are real data.

TABLE 1. 2022 January EUR/USD exchange selling rate.

Date	EUR/USD Exchange Selling
3.01.2022	1,1294
4.01.2022	1,1285
5.01.2022	1,1313
6.01.2022	1,1291
7.01.2022	1,1359
10.01.2022	1,1324
11.01.2022	1,1364
12.01.2022	1,1442
13.01.2022	1,1453
14.01.2022	1,1414

Table 1 contains EUR/USD data for the first 10 working days of January 2022. The data is taken from The Central Bank of The Republic of Türkiye [33]. It will be exemplified how to calculate volatility using these data.

$$\begin{aligned}
P_{\text{total}} &= \sum 1,1294 + 1,1285 \ldots 1,1414 \\
&= 11,3539 \ (P_{\text{total}} \text{ is total value of the all data}) \\
\text{n} &= 10 \ \text{(Total number)} \\
P_{\text{mean}} &= P_{\text{total}}/n = 11,3539/10 = 1,1353 \\
P_{\text{squaresum}} &= \sum (P_{\text{mean}} - Pi)^{2=} 0,00036189 \\
\text{Variance} &= P_{\text{squaresum}}/n = 0,00036189/10 \\
&= 0,000036189 \\
\text{Volatility} &= \sqrt{}\text{Variance} = 0,01902
\end{aligned}$$

The volatility of the 10-day period EUR/USD is 0,01902. An application was developed to make all calculations and other calculations were made through the application.

IV. VERIFICATION OF THE PROPOSED METHOD

In this study, for proof; it is evidenced that derivatives can be taken, there is no idle element in the domain and an element in the domain has two counterparts in the value set.

In addition, it has been proven that the problems existing in other activation functions are not existing in the volatility activation function. These problems are linearity, gradient protection, data convergence, vanishing gradient problems and they are not included in the volatility activation function.

MSE and MAE calculations have been utilized in order to compare the model with literature studies. MSE and MAE are shown in equations 5 and 6.

Mean Squared Error (MSE)

$$MSE = \frac{\sum_{t=1}^{n} (u^2)}{n} \tag{5}$$

Mean Absolute Error (MAE)

$$MAE = \frac{\sum_{t=1}^{n} |u_t|}{n} \tag{6}$$

A. DERIVATIVE

Derivative of the volatility activation function can be taken. The fact that the derivative can be taken is significant in order to learn the model in activation functions. While creating the Artificial Neural Network (ANN) architecture, the activation function is chosen and it is very important that it is differentiable when choosing this activation function. The reason why its derivatization is very important is that it is used to update layers in the ANN [34].

If we look at the derivative of the volatility function without considering the total function, this function includes P_{mean} and P_i variables, n is constant, so there is no need to consider n for derivative.

When defining second and higher order partial derivatives, methods are applied similarly to higher order derivatives of univariate functions. When taking partial derivatives with respect to x and y, first the derivative of f is taken with respect to x, then the derivative of f with respect to y is taken and the result is obtained [37].

The volatility activation function formula is as shown in equation(4). The volatility function consists of n variables. These are x1, x2, ... xn. If it can be partially differentiated with respect to xi, it will be proven that it can be differentiated. Equation 7 shows the volatility activation function. Afterwards, the derivative of the volatility function is shown in equations 8 and 9.

$$f(x_1, \ldots, x_n) = \sqrt{\frac{\sum (\bar{x} - x_i)^2}{n}} = \frac{1}{\sqrt{n}} \cdot \sqrt{\sum (\bar{x} - x_i)^2} \tag{7}$$

Partial derivative of the volatility function with is as shown in equation(8) and equation(9).

$$\frac{\partial f}{\partial x_i} = \frac{1}{\sqrt{n}} \cdot \frac{2.(\bar{x} - x_i)}{2.\sqrt{\sum (\bar{x} - x_i)^2}} \tag{8}$$

$$\frac{\partial f}{\partial x_i} = \frac{1}{\sqrt{n}} \cdot \frac{(\bar{x} - x_i)}{\sqrt{\sum (\bar{x} - x_i)^2}} \tag{9}$$

The partial derivative of the volatility function can be taken according to the variable x_i. In this way, it is seen that the derivative of the volatility activation function can be taken.

B. THE CONDITION OF BEING A FUNCTION ACCORDING TO THE DEFINITION AND VALUE SET

There are rules that a suggestion must follow in order to be a function. These rules are as follows;

- A function must not have any idle elements in its domain.
- Each element in the domain must have only one image in the value set.

To prove that the modified activation function is a function, its proof according to the rules of being a function is as follows. Vertical line test is conducted on the graph to prove these properties.

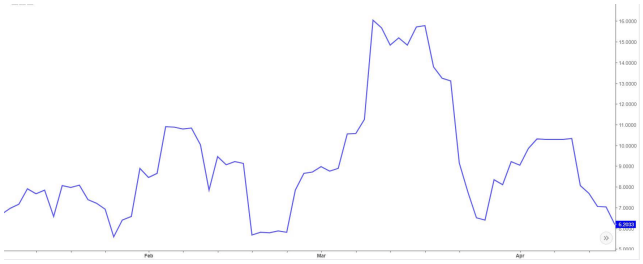

FIGURE 2. 2022 January-April EUR/USD volatility.

There is the volatility graph for 2022 January - April months on Figure 2 [33]. The data is taken from The Central Bank of the Republic of Türkiye. On the graph, vertical linearity test has been conducted for the volatility function with X1, X2 and X3 vertical lines on the graph. According to this test, each value of the volatility function on the x-axis has a corresponding value on the y-axis. For each element on the x-axis, there is only one corresponding in the y-axis, namely in the image set.

TABLE 2. January- april EUR/USD volatility data.

Date	Total Day	First Day Value	Last Day Value	Formula	Volatility Value
2022.01	21	1,129	1,123	f($x^1_{1,\ldots}, x^1_{21}$)	7
2022.02	20	1,126	1,121	f($x^2_{1,\ldots}, x^2_{20}$)	10
2022.03	23	1,112	1,106	f($x^3_{1,\ldots}, x^3_{23}$)	9
2022.04	21	1,105	1,054	f($x^4_{1,\ldots}, x^3_{20}$)	9

Table 2 shows the calculated version of 2022 January-April EUR/USD Volatility values. There are approximately 22 working days per month. Table 2 is a summary table, the original of the table is 88 days. This table contains the EUR/USD value in each month, the volatility value in this month, and the formula used to calculate the volatility value.

C. NON-LINEAR

The backbone of the suggested activation function is the volatility function. The most used activation functions are nonlinear activation functions. The reason for this is that nonlinear activation functions are compatible with the data and

make it easier to distinguish between results [35]. A function must meet two conditions to be linear.

F: $R^n \rightarrow R^m$

1- It must be F(u+v) = F(u)+F(v).
2- It must be F(c.u) = c.F(u) where c is a constant value.

Equations 10, 11, 12, 13 and 14 show the one-month values of the volatility function.

$$F(1) = f(x_1^1, \ldots, x_{21}^1) \quad (10)$$
$$F(2) = f(x_1^2, \ldots, x_{20}^2) \quad (11)$$
$$F(3) = f(x_1^3, \ldots, x_{23}^3) \quad (12)$$
$$F(4) = f(x_1^4, \ldots, x_{20}^3) \quad (13)$$
$$F(i) = f(x_1^i, \ldots, x_{22}^i) \quad (14)$$

There are approximately 22 working days per month. If we look at the limits over the data on the Figure 5,

$F(x_1^1, \ldots, x_{21}^1)$ = January = 7
$F(x_1^2, \ldots, x_{20}^2)$ = February = 10
$F(x_1^3, \ldots, x_{23}^3)$ = March = 9
$F(x_1^4, \ldots, x_{20}^3)$ = April = 9

The data used below comes from Table 2. If it is checked whether the volatility function meets these two conditions:

1- Considering the values on Figure 2 and table 2 F(1) = January = 7, F(2) = February = 10, F(3) = March = 9, F(4) = April = 9.
It must be F(1+2) = F(1) +F(2) to be linear.
F(1+2) = F(3) = 9,
F(1) +F(2) = 7 +10 = 17
17 != 9, so the volatility function is not linear.
2- If the c value is taken as 2, then
F(c.u) = F(2.1) = F(2) = 10.
c.F(u) = 2.F(1) = 2.7 = 14.
14!=10, so the volatility function is not linear.

Year 2020 EUR/USD daily graph and below it, 10-day period volatility graph is given on Figure 3. It is seen on Figure 3 that the volatility is not linear.

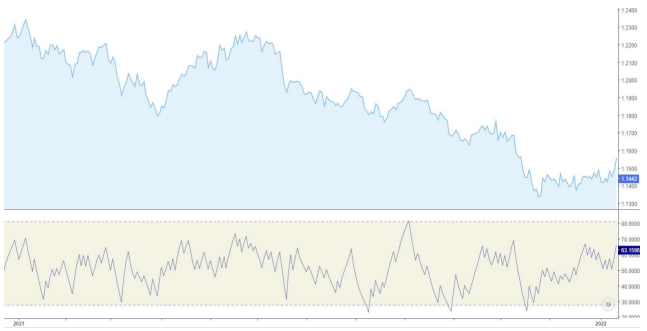

FIGURE 3. 2021 EUR/USD volatility.

A deterministic decision is requested from an artificial neural network in general and this also leads it to be linear. For an X input, non-linearity can be achieved by stochastic adjustment by looking at its scale with other inputs. If we look at the volatility activation function for the proof of non-linearity:

FIGURE 4. EUR/USD daily volatility graph.

There are 2 graphics on the figure 4. The chart at the top is the EUR/USD chart and is the actual data. The graph in the alpha section shows the volatility values. Transactions are made over volatility values.

TABLE 3. EUR/USD 5-day volatility values.

Date	Value
2022.03.07	12.75
2022.03.08	32.00
2022.03.09	44.70
2022.03.10	35.90
2022.03.11	32.06

EUR/USD daily volatility graph is given on Figure 4. [33]. The data is taken from The Central Bank of the Republic of Türkiye. Table 3 contains data calculated according to the volatility formula. After calculating the EUR/USD data according to the volatility formula, the min-max function is applied to keep it within a certain limit range. Since time series data changes according to time, an application has been developed and these calculations are made by application. The volatility data in Table 3 is a small sample data set used to show that the volatility activation function is not linear.

If we prove the non-linearity of the volatility function according to EUR/USD 5-day volatility values on Table 2:

Volatility values for days are expressed as V1, V2, V3, V4, V5. V2-V1 = 32-12,75 = 19,25.

Then, 19,25+V2 = 19,25 + 32 = 51,25.

Is 51,25 equal to v3? 51,25 !=44,70 and this indicates that the volatility function is non-linear.

D. HAVING LIMITS

If an activation function has no limits, under certain conditions, results cannot be obtained. For this reason, it is also important that the activation function should have limits.

The modified activation functions generate values within certain intervals with the min-max method as one of the normalization methods.

According to calculations made for a 10-day period in the graph given on Figure 6, these limits range between 20 - 80. Min –max function is as follows equation 15.

$$S' = \frac{\text{s} - \text{min}}{\text{maks} - \text{min}} \tag{15}$$

Min is the lowest value that can be taken by a data. Max is the highest value that can be taken by a data. S indicates the original value of data while S' indicates the normalized data. The results for volatility function has been reduced to the range of 0-1 with this method. If we calculate for S=60, we have obtained the result (60-20)/(80-20) = 0,66. This result is between 0 and 1.

E. CONTINUITY

A proper activation function must be continuous. Continuity also means sustainability. It is possible to detect continuity and discontinuous points on the graph. If there is no dashed point on the graph, that means the function is continuous. Continuity is one of the most important and fundamental subjects of mathematics [36].

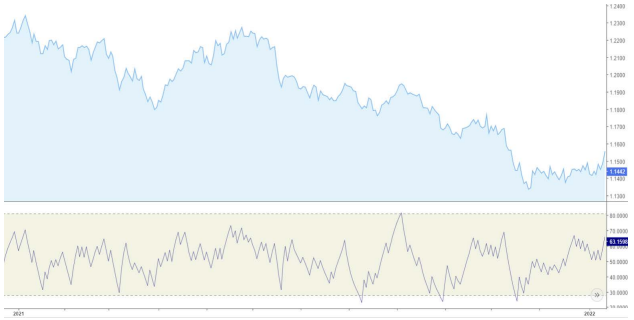

FIGURE 5. 2021 EUR/USD volatility.

The volatility function, which forms the basis of the activation function suggested in Figure 5, is the yellow graph at the bottom. [33]. The data is taken from The Central Bank of the Republic of Türkiye. There is no dashed point on the graph.

Continuity cannot only be detected on the graph, but also by looking at the limit. For the volatility function expressed as F: R →R, if a ∈ R $\lim_{x\to a} f(x) = f(a)$ equality is achieved, f function is called continuous at the point x = a. If the function F is continuous at every point in the domain, then the function f is continuous.

The data are shown in section C and figure 5 also table 2.

$$\lim_{x\to 1} f(x) = 7$$
$$f(1) = 7$$

so the volatility function is continuous at point 1.

$$\lim_{x\to 2} f(x) = 10$$
$$f(2) = 10$$

so the volatility function is continuous at point 2.

Since the volatility function is similarly continuous at other points for the volatility function, it is continuous for all the numbers between [−R, +R].

F. GRADIENT PROTECTION

If we examine if the volatility function has the gradient protection feature;

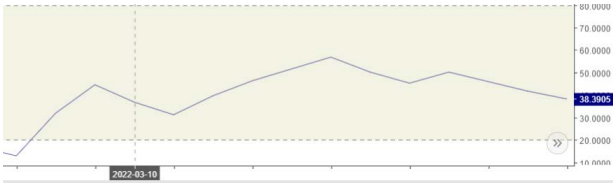

FIGURE 6. 10-day volatility value.

If we examine over the volatility values between 2022-03-07 and 2022-03-25 on Figure 6. [33]. The data is taken from The Central Bank of the Republic of Türkiye. The data in the chart are volatility values calculated from real data.

The derivative of volatility would be;

f' > 0 between 2022-03-07 and 2022-03-09.
f' < 0 between 2022-03-09 and 2022-03-11.
f' > 0 between 2022-03-11 and 2022-03-17.
f' < 0 between 2022-03-17 and 2022-03-21.

The problem of ReLU generating the derivative value of 1 for each case is proved by showing that the derivative values of the volatility function differ between 2022-03-07 and 2022-03-25.

G. VANISHING GRADIENT

Since the Sigmoid and tanh activation functions generate values in the range between [−1,1], their derivatives converge to zero. A 10-day volatility function is given on Figure 6. It has been showed that the function's derivative creates different values for different points. As it can be understood here, the derivative of the volatility function does not converge to zero. For this reason, a solution to the vanishing gradient problem has been generated.

H. DATA CONVERGENCE

Within the scope of this study, the convergence of the activation functions to the time series data has been achieved. The volatility function is used to achieve this convergence. Three different time series data were used for convergence to time series. The test results with these time series are as follows.

V. APPLICATION AND DISCUSSION OF THE PROPOSED METHOD

In this study, the following validity, applicability and proof methods of the suggested activation functions have been implemented in addition to the comparisons with other activation functions.

For validity and applicability; a comparison was made on the same data set with the studies in the literature, validation was made through simulations, accuracy and loss values were compared, and comparisons were made with other activation functions.

TABLE 4. Financial instrument data validation results.

Activation Function	Validation Accuracy	Validation Loss
Sigmoid	0,983	0,0478
ReLU	0,989	0,0453
Volatility	0,994	0,0365

TABLE 5. Time series ETT validation results.

Activation Function	Validation Accuracy	Validation Loss
Sigmoid	0,882	0,2693
ReLU	0,895	0,2512
Volatility	0,923	0,2415

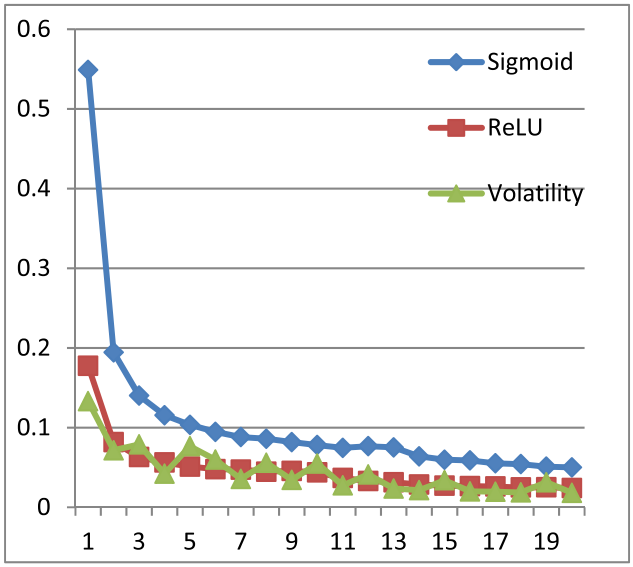

FIGURE 7. Volatility activation function training loss.

A. VOLATILITY ACTIVATION FUNCTION APPLICATION WITH FINANCIAL INSTRUMENT DATA

Ounce Gold is expressed by the symbol OZ, and 1 ounce is equal to 33.1 grams. An application has been developed for the methods suggested within the scope of the study. The test results of the suggested method in the volatility activation function and the commonly used activation functions were calculated with the data of 250 days of ounce gold made over the developed application between the dates of 02-01-2017 and 29-12-2017. [33]. The data is taken from The Central Bank of the Republic of Türkiye.

Table 4 shows commonly used activation functions and the results of activation functions suggested. According to the results, loss values decreased in spite of the increase in validation ratios. The validation ratios of volatility activation function are higher than other activation functions at the ratio of approximately 0.1.

The activation function created in scope of the study and the training loss values of 2 most commonly used activation function until epoch 20 are given on above graph as Figure 7.

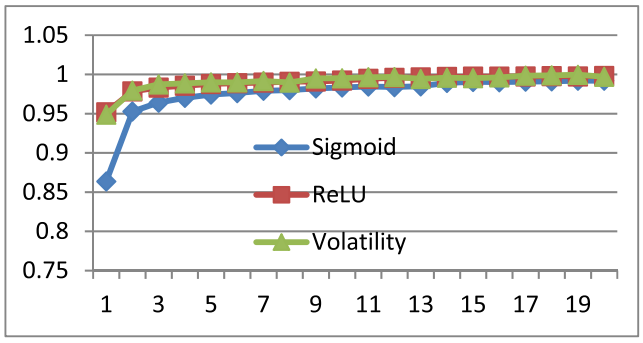

FIGURE 8. Volatility activation functions training accuracy.

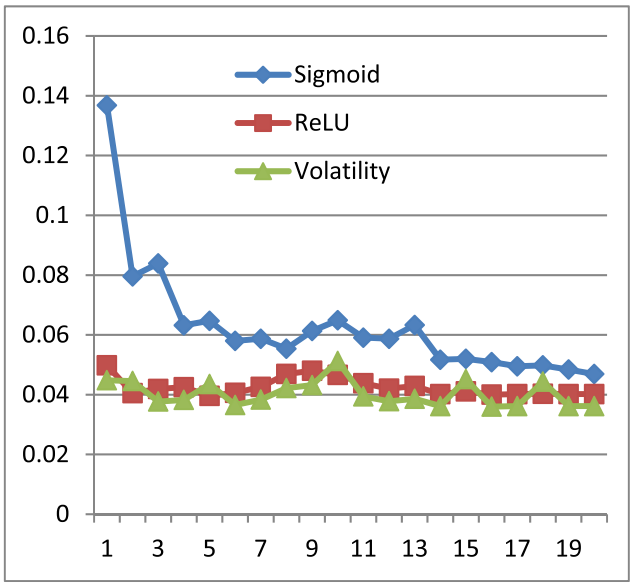

FIGURE 9. Volatility activation functions validation loss.

The activation function created in scope of the study and the training accuracy values of 2 most commonly used activation function until epoch 20 are given on above graph as Figure 8.

The activation function created in scope of the study and the validation loss values of 2 most commonly used activation function until epoch 20 are given on above graph as Figure 9.

The activation function created in scope of the study and the validation accuracy values of 2 most commonly used activation function until epoch 20 are given on above graph as Figure 10.

B. VOLATILITY ACTIVATION FUNCTION APPLICATION WITH ELECTRICAL TRANSFORMER TEMPERATURE DATA

One of the important indicators of electricity distribution is Electrical Transformer Temperature (ETT). A forecasting study was conducted on ETT data obtained from two different districts in China in order to demonstrate the accuracy and applicability of the volatility activation function suggested within the scope of this study. ETTh1 data was used for one time level. These data are oil temperature and load data from July 2016 to July 2018. This data includes data recorded

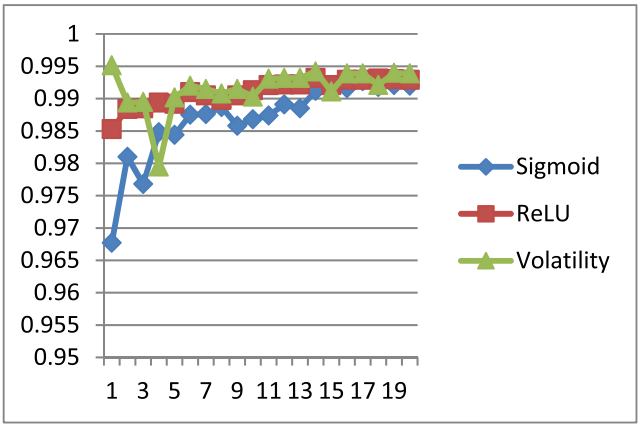

FIGURE 10. Volatility activation function validation accuracy.

TABLE 6. Time series rain forcast results.

Activation Function	Validation Accuracy	Validation Loss
Sigmoid	0,815	0,367
ReLU	0,821	0,356
Volatility	0,852	0,336

every 15 minutes [21]. Wu et al. [22] studied on long-term forecasting by utilizing ETT data in his study.

The results of the study conducted for forecasting temperature with ETT data are given on Table 4. For the suggested method, the following results have been found: MSE = 0,362 and MAE = 0,448. The results of this study conducted with the suggested method have revealed similar results with the studies in the literature [23], [24], [25], [26].

C. NEXT DAY RAIN FORECAST EXAMPLE IN AUSTRALIA

In order to determine the suitability of the proposed methods, the accuracy rates of the studies applied to the time series problem in the literature were compared. The next day's rain forecast example in Australia was examined for the weather forecast [32].

A weather forecast is a prediction of what the weather will be like in the future based on past weather data. Air pressure, humidity, wind, temperature and other measurements are used by meteorologists and meteorology applications and can be used in many formulas. Sharp observation skills require weather formulas and lots of weather data for highly accurate weather forecasting. We can make forecasts for the coming days by identifying patterns in the data with trained deep learning models.

Table 6 contains the results of the study on rain forecasting. Literature study results are val_loss: 0.3563 - val_accuracy: 0.8425. After applying the proposed method to the same test data, an increase in accuracy rates of 0.2 was observed. The accuracy value increased by 0.3 compared to other activation functions.

VI. CONCLUSION

The problems of data convergence, non-linearity, gradient protection, vanishing gradient are ongoing in ReLU, Sigmoid and the activation functions generated on these activation functions. Even though there are many studies in the literature for the solution of these problems existing in ReLU and Sigmoid activation functions, any of them did not take over the position of the ReLU and sigmoid activation functions. Not only some contributions have been provided for the literature but also accuracy ratios have been increased by suggesting and presenting alternative solutions for the problems of data convergence, non-linearity, gradient protection, vanishing gradient with the volatility activation function. Some solutions have been presented for all these 4 problems concerned in several studies in the literature by means of this study. Since only single problem is focused in most studies in the literature, an inclusive architecture was not created yet. In this study, a suggestion including all these problems is set forth unlike the previous literature studies.

Time series forecasting is performed with the volatility activation function within the scope of the study. Financial instruments data, ounce gold data and electrical transformer temperature data have been utilized as the time series data. According to the results of the tests conducted with ounce gold data, some increase occurred on a ratio of 0.01 in accuracy. As a result of the test studies conducted with electrical transformer data, some MAE and MSE data similar with the literature have been obtained. Characteristics of the volatility activation function are as follows: its derivative can be taken, there is no idle element in the domain and each element in the domain has a counterpart in the value set, it is non-linear, it has limits, it is continuous, does not have the problem of gradient protection, does not have the problem of vanishing gradient, and has the characteristic of data convergence.

Another problem in the literature studies is the fact that suggested methods are not easily understandable and applicable like Sigmoid, ReLU. For that reason, newly conducted studies fail in replacing the Sigmoid and ReLU activation functions. On the other hand, the volatility activation function may be an alternative for ReLU and Sigmoid functions because it is easily understandable and applicable. In conclusion of the study, it has been found that the volatility activation function is an applicable, proved and strong alternative activation function candidate.

With the volatility activation function proposed within the scope of the study, a new activation function has been introduced to the literature. However, the volatility activation function produced higher accuracy results than the most widely used ReLU and Sigmoid functions in the literature. In addition, in the applicability studies conducted with three different data sets, better results were obtained than the literature studies.

ACKNOWLEDGMENT

The authors would like to thank the journal editors and referees for their valuable time and ideas for the article to be of better quality and effectiveness. The article is derived from Furkan Kayim's doctoral thesis of the same name, supervised by Atinç Yilmaz.

REFERENCES

[1] S. Kiliçarslan, K. Adem, and M. Çelik, "An overview of the activation functions used in deep learning algorithms," *J. New Results Sci.*, vol. 10, no. 3, pp. 75–88, Dec. 2021.
[2] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.
[3] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[4] E. S. de Oliveira Marques, K. S. T. R. Alves, D. Pekaslan, and E. P. Aguiar, "Kernel evolving participatory fuzzy modeling for time series forecasting: New perspectives based on similarity measures," in *Proc. IEEE Int. Conf. Evolving Adapt. Intell. Syst. (EAIS)*, May 2022, pp. 1–8.
[5] Y.-Y. Fanjiang, Y. Syu, and W.-L. Huang, "Time series QoS forecasting for web services using multi-predictor-based genetic programming," *IEEE Trans. Services Comput.*, vol. 15, no. 3, pp. 1423–1435, May 2022.
[6] S. J. Basha, T. Ammannamma, K. Vivek, and V. S. Veesam, "Comparative analysis of time series forecasting models to predict amount of rainfall in Telangana," in *Proc. 8th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Mar. 2022, pp. 1918–1922.
[7] N. G. El-Din Saad, S. Ghoniemy, H. Faheem, and N. A. Seada, "An evaluation of time series-based modeling and forecasting of infectious diseases progression using statistical versus compartmental methods," in *Proc. 5th Int. Conf. Comput. Informat. (ICCI)*, Mar. 2022, pp. 263–273.
[8] G. Liu, "Time series forecasting via learning convolutionally low-rank models," *IEEE Trans. Inf. Theory*, vol. 68, no. 5, pp. 3362–3380, May 2022.
[9] R. Gupta, A. K. Yadav, S. Jha, and P. K. Pathak, "Time series forecasting of solar power generation using Facebook prophet and XG boost," in *Proc. IEEE Delhi Sect. Conf. (DELCON)*, Feb. 2022, pp. 1–5.
[10] P. Malhan and M. Mittal, "Comparison of decomposition-based machine learning models for multi-step time series forecasting of wind power generation," in *Proc. IEEE 2nd Int. Conf. Electr. Power Energy Syst. (ICEPES)*, Dec. 2021, pp. 1–6.
[11] M. Maya, W. Yu, and X. Li, "Time series forecasting with missing data using neural network and meta-transfer learning," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2021, pp. 1–6.
[12] L. A. Demidova and A. V. Gorchakov, "A study of biology-inspired algorithms applied to long short-term memory network training for time series forecasting," in *Proc. 3rd Int. Conf. Control Syst., Math. Modeling, Autom. Energy Efficiency (SUMMA)*, Nov. 2021, pp. 473–478.
[13] S. Sarikaa and S. Niranjana, "Time series forecasting of cloud resource usage," in *Proc. IEEE 6th Int. Conf. Comput., Commun. Autom. (ICCCA)*, Dec. 2021, pp. 372–382.
[14] E. Mouine, Y. Liu, J. Sun, M. Nayrolles, and M. Kalantari, "The analysis of time series forecasting on resource provision of cloud-based game servers," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2021, pp. 2381–2389.
[15] M. Carpenter, C. Luo, and X.-S. Wang, "The effects of regularisation on RNN models for COVID-19 time series forecasting," in *Proc. 20th Int. Conf. Ubiquitous Comput. Commun. (IUCC/CIT/DSCI/SmartCNS)*, Dec. 2021, pp. 281–287.
[16] Y. Lin, I. Koprinska, and M. Rana, "SSDNet: State space decomposition neural network for time series forecasting," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2021, pp. 370–378.
[17] G. Zuniga and G. Acuna, "Probabilistic multistep time series forecasting using conditional generative adversarial networks," in *Proc. IEEE Latin Amer. Conf. Comput. Intell. (LA-CCI)*, Nov. 2021, pp. 1–6.
[18] C. Kellermann, E. Neumann, and J. Ostermann, "A new preprocessing approach to reduce computational complexity for time series forecasting with neuronal networks: Temporal resolution warping," in *Proc. Int. Symp. Comput. Sci. Intell. Controls (ISCSIC)*, Nov. 2021, pp. 324–328.
[19] B. Eua-Arporn, S.-L. Huang, and E. E. Kuruoglu, "Enhancing neural network based hybrid learning with empirical wavelet transform for time series forecasting," in *Proc. IEEE 33rd Int. Conf. Tools With Artif. Intell. (ICTAI)*, Nov. 2021, pp. 386–390.
[20] D. Jacoby, J. Ostrometzky, and H. Messer, "Adaptive fuzzy-based models for attenuation time series forecasting," in *Proc. IEEE Int. Conf. Microw., Antennas, Commun. Electron. Syst. (COMCAS)*, Nov. 2021, pp. 522–528.
[21] K. K. R. Samal, K. S. Babu, and S. K. Das, "Time series forecasting of air pollution using deep neural network with multi-output learning," in *Proc. IEEE 18th India Council Int. Conf. (INDICON)*, Dec. 2021, pp. 1–5.
[22] J.-H. Choi, J.-Y. Yang, and J.-H. Chang, "Short-utterance embedding enhancement method based on time series forecasting technique for text-independent speaker verification," in *Proc. IEEE Autom. Speech Recognit. Understand. Workshop (ASRU)*, Dec. 2021, pp. 130–137.
[23] Y. Liao and C. Liang, "A temperature time series forecasting model based on DeepAR," in *Proc. 7th Int. Conf. Comput. Commun. (ICCC)*, Dec. 2021, pp. 1588–1593.
[24] T. Moposita, L. Trojman, F. Crupi, M. Lanuzza, and A. Vladimirescu, "Voltage-to-voltage sigmoid neuron activation function design for artificial neural networks," in *Proc. IEEE 13th Latin Amer. Symp. Circuits Syst. (LASCAS)*, Mar. 2022, pp. 1–4.
[25] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI*, 2021, pp. 11106–11115.
[26] H. Wu. (2021). *Autoformer: Decomposition Transformers With Auto-Correlation for Long-Term Series Forecasting*. [Online]. Available: https://paperswithcode.com/paper/autoformer-decomposition-transformers-with
[27] M. Liu, A. Zeng, Z. Xu, Q. Lai, and Q. Xu, *Time Series is a Special Sequence: Forecasting With Sample Convolution and Interaction*. Hong Kong: Chinese Univ. Hong Kong, 2021.
[28] H. Wu, "Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting," in *Proc. 35th Conf. Neural Inf. Process. Syst.* Beijing, China: Tsinghua Univ., China, School of Software, BNRist, 2022.
[29] A. Zeng, *Are Transformers Effective for Time Series Forecasting*. Beijing, China: Chinese Univ. Hong Kong, 2022.
[30] S. Liu, "Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting," in *Proc. Conf. Paper ICLR*, Sep. 2022, pp. 1–3.
[31] H. Zhou, *Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting*. Beijing, China: Beihang Univ., 2020.
[32] J. Young. (2021). *Predict Next-Day Rain in Australia*. [Online]. Available: https://www.kaggle.com/jsphyg/weather-dataset-rattle-package
[33] (Mar. 2022). *TCMB*. [Online]. Available: https://evds2.tcmb.gov.tr/index.php?/evds/serieMarket
[34] L. Alfonso. (Apr. 2022). *Why do we use the Derivatives of Activation Functions in a Neural Network*. [Online]. Available: https://medium.com/@alfonsollanes/why-do-we-use-the-derivatives-of-activation-functions-in-a-neural-network
[35] L. Panneerselvam, "Activation functions and their derivatives—A quick & complete guide," in *Data Science Blogathon*, 2021, p. 5. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/04/activation-functions-and-their-derivatives-a-quick-complete-guide/
[36] D. S. Memnun, F. Sevindik, C. Beklen, and E. Dinç, "Analysis of the abstraction process of continuity knowledge," *World J. Educ.*, vol. 9, no. 2, p. 141, Apr. 2019.
[37] A. C. Chiang, *Fundamental Methods of Mathematical Economics*, 3rd ed. New York, NY, USA: McGraw-Hill, 1984, pp. 316–318.

FURKAN KAYIM was born in Yozgat, in 1988. He received the undergraduate degree in computer-related applications and educational sciences from the Computer Engineering Department, Selçuk University, in 2007, and the master's degree from the Department of Computer Engineering, Institute of Science and Technology, Beykent University, in 2015, where he is currently pursuing the Ph.D. degree with the Department of Computer Engineering. In 2012, he joined a private company in the IT sector. He has been working in the sector for ten years as a Software Specialist.

ATINÇ YILMAZ was born in Izmit, in 1983. He received the undergraduate and graduate degrees from the Department of Computer Engineering and the Ph.D. degree from the Department of Computer and Informatics Engineering, Sakarya University.

He has been an academician, since 2005, and has given lectures on computer engineering and artificial intelligence in different universities. He is a Faculty Member with the Department of Computer Engineering, Faculty of Engineering and Architecture, Beykent University, where he is the Head of the Department of Computer Engineering.