frontiers | Frontiers in Neurorobotics

TYPE Original Research
PUBLISHED 23 September 2022
DOI 10.3389/fnbot.2022.1022887

A novel activation function based recurrent neural networks and their applications on sentiment classification and dynamic problems solving

Qingyi Zhu[1]† and Mingtao Tan[2]*†

[1]School of Electronics and Internet of Things, Sichuan Vocational College of Information Technology, Guangyuan, China, [2]School of Computer and Electrical Engineering, Hunan University of Arts and Science, Changde, China

OPEN ACCESS

EDITED BY
Tiantai Deng,
The University of Sheffield,
United Kingdom

REVIEWED BY
Fei Yu,
Changsha University of Science and Technology, China
Long Chen,
University of Leicester,
United Kingdom
Jie Jin,
Hunan University of Science and Technology, China
Cheng Yin,
University of Surrey, United Kingdom

*CORRESPONDENCE
Mingtao Tan
scuzqy7624@163.com

†These authors have contributed equally to this work

RECEIVED 19 August 2022
ACCEPTED 31 August 2022
PUBLISHED 23 September 2022

CITATION
Zhu Q and Tan M (2022) A novel activation function based recurrent neural networks and their applications on sentiment classification and dynamic problems solving.
Front. Neurorobot. 16:1022887.
doi: 10.3389/fnbot.2022.1022887

COPYRIGHT
© 2022 Zhu and Tan. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

In this paper, a nonlinear activation function (NAF) is proposed to constructed three recurrent neural network (RNN) models (Simple RNN (SRNN) model, Long Short-term Memory (LSTM) model and Gated Recurrent Unit (GRU) model) for sentiment classification. The Internet Movie Database (IMDB) sentiment classification experiment results demonstrate that the three RNN models using the NAF achieve better accuracy and lower loss values compared with other commonly used activation functions (AF), such as ReLU, SELU etc. Moreover, in terms of dynamic problems solving, a fixed-time convergent recurrent neural network (FTCRNN) model with the NAF is constructed. Additionally, the fixed-time convergence property of the FTCRNN model is strictly validated and the upper bound convergence time formula of the FTCRNN model is obtained. Furthermore, the numerical simulation results of dynamic Sylvester equation (DSE) solving using the FTCRNN model indicate that the neural state solutions of the FTCRNN model quickly converge to the theoretical solutions of DSE problems whether there are noises or not. Ultimately, the FTCRNN model is also utilized to realize trajectory tracking of robot manipulator and electric circuit currents computation for the further validation of its accurateness and robustness, and the corresponding results further validate its superior performance and widespread applicability.

KEYWORDS

non-linear activation function, recurrent neural networks, sentiment classification, dynamic sylvester equation, robot manipulator

Introduction

Sentiment classification and time-varying problem solving are two typical problems in practical applications. Sentiment classification utilizes computational techniques and natural language processing to classify specific text into positive and negative categories (Abdi et al., 2018; Chen and Xie, 2020; Wang and Lin, 2020; Saeed et al., 2021). As the document information generated by users increasing rapidly, the analysis of this information becomes more and more important. Based on the analysis of this information, it can obtain the feedback information from students, products, services and events, which is useful for decision making of universities, companies and governments (Yaslan and Aldo, 2017). As a powerful tool for calculation and natural language processing, the recurrent neural network (RNN) is widely used in medical treatment (He et al., 2020), sentiment classification (Azadeh et al., 2017; Zablith and Osman, 2019) and other fields (Yu et al., 2019, 2022a,b; Wan et al., 2022a,b). Note that, the performance of the RNN models for sentiment classification depends on its activation function (AF), which implies that the selection of AF will affect the accuracy of sentiment classification. Therefore, several activation functions are designed, for example, Rectified Linear Unit (ReLU), Leaky Rectified Linear Unit (LReLU), Exponential Linear Unit (ELU) and Scaled ELU (SELU). Motivated by the above discussions, a non-linear activation function (NAF) is proposed in this work. The proposed NAF is applied to a two-layer Simple RNN (SRNN) model, a Long Short-term Memory (LSTM) model and a Gated Recurrent Unit (GRU) model for sentiment classification, respectively. For the purpose of comparison, the RNN models (SRNN, LSTM, and GRU) activated by other commonly used AFs are also used for the same sentiment classification task. The simulation results demonstrate that the three NAF-based RNN models have better accuracy and faster loss decline than the RNN models activated by other commonly used AFs.

On the other hand, solving time-varying problems is becoming increasingly important in the fields of science and engineering (Gong and Jin, 2021; Jin and Gong, 2021; Liu et al., 2021; Jin and Qiu, 2022; Jin et al., 2022a,b,c,d; Zhu et al., 2022). As most time-varying problems can be described by time-varying matrix equations, the RNN is also widely used to effectively solving them in the past few years (Zhang et al., 2002; Xiao and Zhang, 2014b; Xiao et al., 2019; Jin, 2021a; Liu et al., 2022). Zeroing neural network (ZNN) is a typical RNN model developing rapidly for time-varying matrix equations solving, and many researchers devote to improve the convergence and robustness of the ZNN model in recent years. For example, the varying-parameter ZNN model is proposed by Zhang et al. (2018a,b,c, 2020b,d). The varying-parameter ZNN model is a parallel processing approach with high-efficiency and high-precision, and its unique advantage is that it is a real-time solver without any pre-training. Moreover, a large number of relevant literatures reveal that the performances of ZNN model are intrinsically related to its AF. Therefore, researchers have designed many AFs such as LAF and PSAF, and the ZNN model activated by them could exponentially converge to the theoretical solutions of dynamic problems in ideal no noise environment (Zhang et al., 2018c). Besides, the SBPAF activated ZNN model even converges in finite time, and the RNZNN models in ref. (Li et al., 2013) achieve fixed-time convergence and strong robustness to noises. Furthermore, the varying-parameter ZNN model achieves super-exponential convergence and strong robustness by introducing a variable convergent factor γ (Zhang et al., 2020c; Xiao and He, 2021). Although the improvements of AF and convergent factor γ both lead to the improved performances of the ZNN model, this work focuses on the development of AF of the ZNN model, and a NAF is presented and employed to obtain a fixed-time convergent recurrent neural network (FTCRNN) model for further enhance its convergence and robustness, and the FTCRNN model not only has strong robustness to noises but also converges in fixed-time for time-varying problems solving.

Consequently, the main contributions and innovations of this work are summarized as follows:

(1) An NAF suits for different RNN models is proposed.
(2) Three NAF activated RNN models (SRNN, LSTM, and GRU) are developed for sentiment classifications.
(3) A robust and fixed-time convergent FTCRNN model for DSE problem solving and robot manipulator trajectory tracking is presented.
(4) An application of electric circuit currents calculation based on the FTCRNN model is designed for its further validation.

The rest of this paper is organized as follows. The introduction of the proposed NAF is presented in Section The proposed NAF. In Section NAF-Based RNN models for Sentiment classification, the sentiment classification problem is formulated and the experiments of sentiment classification are presented. In Section NAF-Based FTCRNN model for dynamic problems solving, the effectiveness and robustness of the NAF-based FTCRNN model are verified by mathematical analysis. Besides, the experiments of time-varying DSE problem solving, robot manipulator trajectory tracking and electric circuit currents calculation using the NAF-based FTCRNN model are provided. Ultimately, the conclusions and future research directions are discussed in Section Conclusions.

The proposed NAF

AF is an important part of RNN, which has great influences on the performances of RNN models. For example, linear AF-based RNN models are suitable for linear problems solving, and non-linear AF-based RNN models can effectively solve the non-linear problems while the linear AF-based RNN models are powerless in non-linear problems.

The four basic AFs for RNN models to realize sentiment classification are presented in Table 1. The ReLU-based RNN models have the advantage of unnecessary pre-processing operation. However, ReLU is relatively sparse, the useful information of the ReLU-based RNN models is easy to be ignored. Therefore, various improved AFs are proposed [e.g., Leak ReLU (LReLU), Exponential Linear Unit (ELU), Scaled ELU (SELU)…].

TABLE 1 Recently reported AFs for sentiment classification.

No.	AFs	Expression
1	ReLU	$\phi(x) = max(0, x)$
2	LReLU	$\phi(x) = \begin{cases} x, & x > 0 \\ \alpha x, & otherwise \end{cases}$
3	ELU	$\phi(x) = \begin{cases} x, & x > 0 \\ \alpha(e^x - 1), & otherwise \end{cases}$
4	SELU	$\phi(x) = \begin{cases} x, & x > 0 \\ \alpha e^x - \alpha, & otherwise \end{cases}$
5	Tanh	$\phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

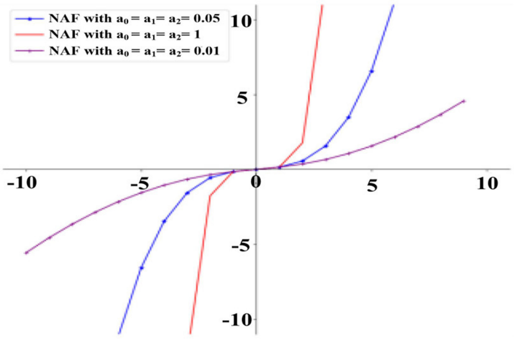

FIGURE 1
NAF (1) with different parameters.

The convergence characteristics of RNN models are closely related to the slope of their AFs (Xiao et al., 2018). The adjustable slope AF based RNN models have better flexibility in various problems solving, and their slope can be set according to specific practical requirements. Obviously, when $x > 0$, the slopes of AFs in Table 1 are constant and they are not adjustable. Therefore, in order to further improve the performance of the RNN models for sentiment classification and time-varying problems solving, and inspired by the method in Chen et al. (2020), a new NAF with flexible slope adjustment property is proposed.

$$\phi(x) = \alpha_0 x + \alpha_1 sign(x)|x|^{2\phi_1 - 1} + \alpha_2 sign(x)|x|^{2\phi_2 - 1} \qquad (1)$$

where $\alpha_0 > 0, \alpha_1 > 0, \alpha_2 > 0, \phi_1 > 1, 0 < \phi_2 < 1$, and $sign\ (\bullet)$ is the signum function.

In order to select proper parameter values of NAF (1) for sentiment classification, the following two-step method is applied.

Firstly, for the convenience of observation, the proposed NAF with different parameters ($a_0, \alpha_1, and\ \alpha_2$) are plotted from the interval $[-10, 10]$ in Figure 1. As seen in Figure 1, the slope of the NAF is closely related to the above parameters, and the middle blue curve with $\alpha_0 = \alpha_1 = \alpha_2 = 0.05$ is adopted.

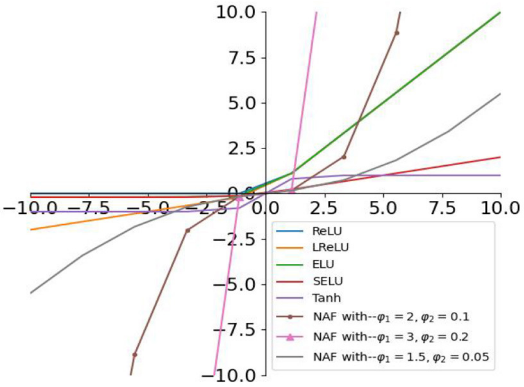

FIGURE 2
NAF (1) and AFs in Table 1.

Then, the AFs in Table 1 and the proposed NAF are plotted in Figure 2. As observed in Figure 2, the slope of the proposed NAF can also be adjusted by the parameters ϕ_1 and ϕ_2, and the middle curve with $\phi_1 = 1.5, \phi_2 = 0.1$ is adopted.

Generally, the parameters in Equation (1) can be chosen arbitrarily as long as they satisfy $\alpha_0 > 0, \alpha_1 > 0, \alpha_2 > 0, \phi_1 > 1, 0 < \phi_2 < 1$. Here, we set $\alpha_0 = \alpha_1 = \alpha_2 = 0.05, \phi_1 = 1.5, \phi_2 = 0.1$ to ensure the NAF-based RNN models for sentiment classification achieve moderate convergence and robustness.

NAF-based RNN models for sentiment classification

Sentiment classification is to judge whether the text information is positive or negative (e.g., good or bad, like or hate), which gathers classification information for users to make decision. In recent years, sentiment classification has been widely used in business, politics, social media, and other fields. In this section, three NAF activated RNN models (SRNN, LSTM and GRU) will be used for Internet Movie Database (IMDB) sentiment classification. Besides, the RNN models (SRNN, LSTM, and GRU) activated by other commonly used AFs in Table 1 are also used for IMDB sentiment classification for the purpose of comparison.

Headings sentiment classification diagram

In this subsection, the diagram for IMDB sentiment classification is introduced. As shown in Figure 3, sentiment classification mainly includes the pre-processing part and the deep learning sentiment classification part.

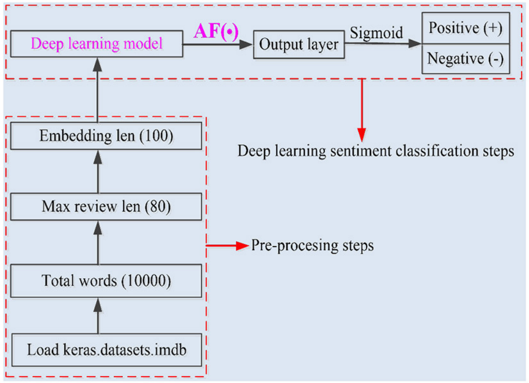

FIGURE 3
Sentiment classification diagram.

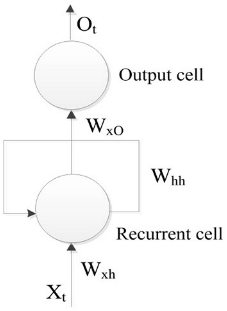

FIGURE 4
SRNN model.

The IMDB is used for sentiment analysis in this work, and the IMDB consists of 50,000 movie reviews. Before the training of the RNN models, the pre-processing steps are presented below.

Firstly, 25,000 reviews of the IMDB are used for test set, and the rest of 25,000 reviews are used for training set;

Secondly, the English vocabulary of the RNN models is set to be 10,000;

Thirdly, only 100 words of each comment can be read by the RNN models;

Finally, each English word is transformed to be a 100-dimensional vector.

After the pre-processing steps, the commonly used AF-based RNN models and the proposed NAF-based RNN models are used for sentiment classification, respectively.

It is worth pointing out that we use the command "word2vec" in Python to convert IMDB into vectors for calculation and training. "Word2vec" is an efficient model for training word vectors. It converts text data into low dimensional real number vectors through unsupervised training.

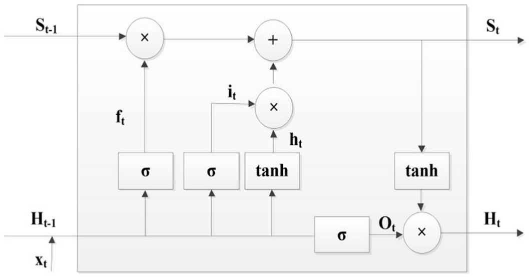

FIGURE 5
LSTM model.

Deep learning models for sentiment classification

The SRNN, LSTM and GRU models are used as deep learning models for sentiment classification in this work, and the corresponding introduction of the three RNN models are presented below.

SRNN model

The SRNN model is shown in Figure 4, and its operation can be expressed mathematically as

$$\begin{aligned} H_t &= f(W_{xh}X_t + W_{hh}H_{t-1}) \\ O_t &= g(W_{xO}H_t) \end{aligned} \tag{2}$$

where X_t is input at time. H_t is state at time. W_{xh}, W_{hh} and W_{xO} are weight matrices.

LSTM model

Generally, the LSTM model consist of memory cell, input gate, output gate and forgetting gate, and the LSTM model is presented in Figure 5.

As a typical RNN model, the LSTM model has long-term or short-term memory, and the results of each LSTM cell are related to its current and past states. The LSTM model effectively solves the problems of gradient vanishing and gradient explosion, and its mathematical expression is presented below.

$$\begin{aligned} &f_t = \sigma(W_f(h_{t-1}, x_t) + b_f) \quad && i_t = \sigma(W_i(h_{t-1}, x_t) + b_i) \\ &O_t = \sigma(W_o(h_{t-1}, x_t) + b_o) \quad && h_t = \tanh(x_t, H_{t-1}) \\ &S_t = S_{t-1}f_t + h_t i_t \quad && H_t = \tanh(S_t) \times O_t \end{aligned} \tag{3}$$

GRU model

The GRU model is the simplified version of the LSTM model, and it can also solve the problems of gradient vanishing and gradient explosion. Compared with the LSTM model, the GRU

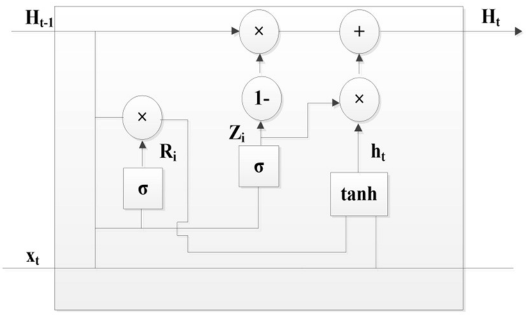

FIGURE 6
GRU model.

TABLE 2 Computer hardware and software.

CPU	Inter i5-7400
GPU	Inter HD Graphics 630
RAM	8 GB
Python version	3.0
tensorflow version	2.0

model only has the update gate and reset gate, and it is easier to be trained.

The GRU model is represented in Figure 6, and its mathematical expression is

$$\begin{aligned} Z_t &= \sigma(W_z(H_{t-1}, x_t)) \ \ R_t = \sigma(W_r(H_{t-1}, x_t)) \\ h_t &= \tanh(R_t h_{t-1}, x_t) \ \ H_t = (Z_t \times h_t) + ((1 - Z_t) \times H_{t-1}) \end{aligned} \tag{4}$$

IMDB sentiment classification

In this subsection, the IMDB sentiment classification experiments are presented to verify the superior performances of the proposed NAF. The hardware and software of the computer used for sentiment classification are presented in Table 2. The three RNN models all use the categorical cross entropy loss function, and their training batch size values are 128 and 30 epochs. Moreover, the number of neurons in the SRNN, LSTM, and GRU models is set as 100 in the input layer, 64 in the hidden layer, 64 in the full connection layer and 1 in the output layer.

Firstly, the SRNN model activated by the proposed NAF and other recently reported AFs in Table 1 are used for the IMDB sentiment classification, and the experiment results are presented in Table 3 and Figure 7. As observed in Table 3, the NAF-based SRNN model and the SRNN model activated by other recently reported AFs all have good training effects with training accuracy >96.5% and training loss <0.93. However, the NAF-based SRNN model is the best one among them, and its training accuracy and training loss achieves 97.2700% and 0.0793, respectively. Moreover, we can also observe that the NAF-based SRNN model and the SRNN model activated by other recently reported AFs all have good sentiment classification effects with sentiment classification accuracy >83.62% and sentiment classification loss <0.9275. Furthermore, the sentiment classification accuracy of the NAF-based SRNN model even achieves 83.9000%, its loss value is 0.4473, which is the lowest of all the SRNN models.

TABLE 3 SRNN model with different AFs for IMDB sentiment classification.

No.	AFs	Training accuracy (%)	Training loss	Val-accuracy (%)	Val-loss
1	ReLU	96.7400	0.0949	83.6300	0.9275
2	LReLU	96.9000	0.0911	83.3200	0.7104
3	ELU	97.0000	0.0864	83.6200	0.9135
4	SELU	96.6600	0.0957	83.0600	0.8323
5	Tanh	96.5700	0.1010	83.4700	0.9083
6	NAF	97.2700	0.0793	83.9000	0.4473

In order to further clearly illustrate the training and sentiment classification effects of the SRNN model activated by the five AFs, the data in Table 3 is plotted in Figure 7. Figures 7A,B are the training curves and the tested sentiment classification curves, respectively. The X-axis represents the number of iterations, and the Y-axis represents the accuracy and loss values. As observed in Figure 7A, the training accuracy of the NAF-based SRNN model is clearly higher than other SRNN models, and the training loss value is always smaller than other SRNN models. In addition, as observed in Figure 7B, the test loss value of the NAF-based SRNN model is extraordinary stable comparing with other SRNN models.

Secondly, the LSTM model activated by the proposed NAF and other recently reported AFs in Table 1 are also used for the IMDB sentiment classification, and the corresponding experiment results are presented in Table 4 and Figure 8. From Table 4 and Figure 8, it is clear that the sentiment classification accuracy of the LSTM model activated by the proposed NAF achieves 83.8700%, and its loss value is also the best one among them.

Thirdly, the IMDB sentiment classification is realized by the GRU model activated by the proposed NAF and other Afs in Table 1, and the corresponding results are presented in Table 5 and Figure 9. As seen in Table 5 and Figure 9, the sentiment classification loss value of the GRU model activated by the proposed NAF is the lowest than other models, and it achieves best sentiment classification accuracy with fewer epochs among all the models.

Finally, in order to illustrate the training process of SRNN, LSTM and GRU models for sentiment classification more

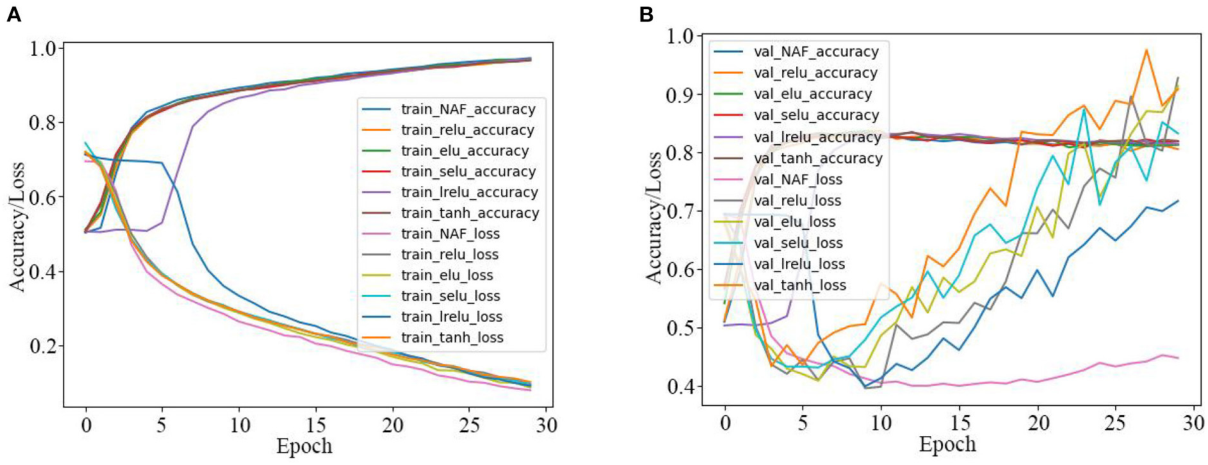

FIGURE 7
Training and test curves of SRNN model for IMDB sentiment classification. **(A)** Training curves. **(B)** Tested sentiment classification curves.

TABLE 4 LSTM model with different AFs for IMDB sentiment classification.

No.	AFs	Train accuracy (%)	Train loss	Val-accuracy (%)	Val-loss
1	ReLU	95.5400	0.1227	83.6900	0.6315
2	LReLU	95.6000	0.1222	83.4200	0.6693
3	ELU	96.0300	0.1103	83.8900	0.7652
4	SELU	95.9400	0.1137	83.8900	0.8349
5	Tanh	95.8100	0.1189	83.6800	0.7827
6	NAF	95.7600	0.1181	83.8700	0.6226

clearly, the output layer weights of three models are presented in Figure 10.

NAF-based FTCRNN model for dynamic problems solving

Sylvester equation is frequently applied in science and engineering fields, and many problems can be solved by finding the solution of Sylvester equations (Darouach, 2006; Zhang et al., 2018c).

In this section, for the purpose of verifying the superior performance of the FTCRNN model, preliminary mathematical preparation and theoretical analysis are given. Besides, the NAF-based FTCRNN model is used for three practical dynamic problems solving, which are dynamic Sylvester equation solving, robot manipulator trajectory tracking and electric circuit currents calculation. Moreover, the original ZNN activated by other recently reported AFs in Table 1 are also used to solve the above three practical dynamic problems for the purpose of comparison.

The dynamic sylvester equation

The DSE problem can be summarized in the following matrix equation.

$$A(t)X(t) - X(t)B(t) = -C(t) \in \mathbb{R}^{n\times n} \tag{5}$$

where t represents time, $X(t) \in \mathbb{R}^{n\times n}$ is the unknown matrix to be solved. $A(t) \in \mathbb{R}^{n\times n}$, $B(t) \in \mathbb{R}^{n\times n}$ and $C(t) \in \mathbb{R}^{n\times n}$ are the known smooth dynamic coefficient matrices, and their time derivatives matrices are also assumed to be known.

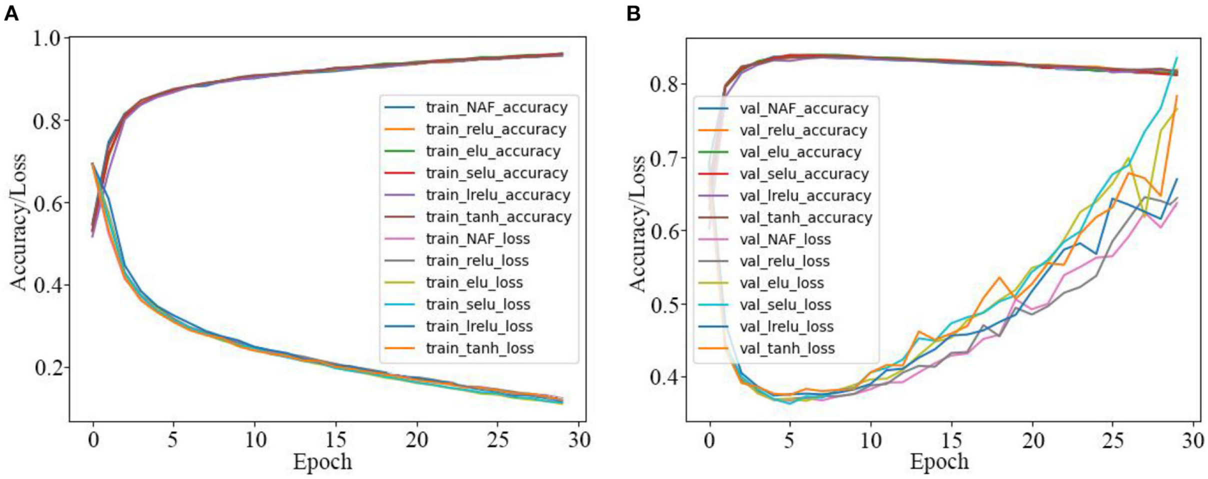

FIGURE 8
Training and test curves of LSTM model for IMDB sentiment classification. **(A)** Training curves. **(B)** Tested sentiment classification curves.

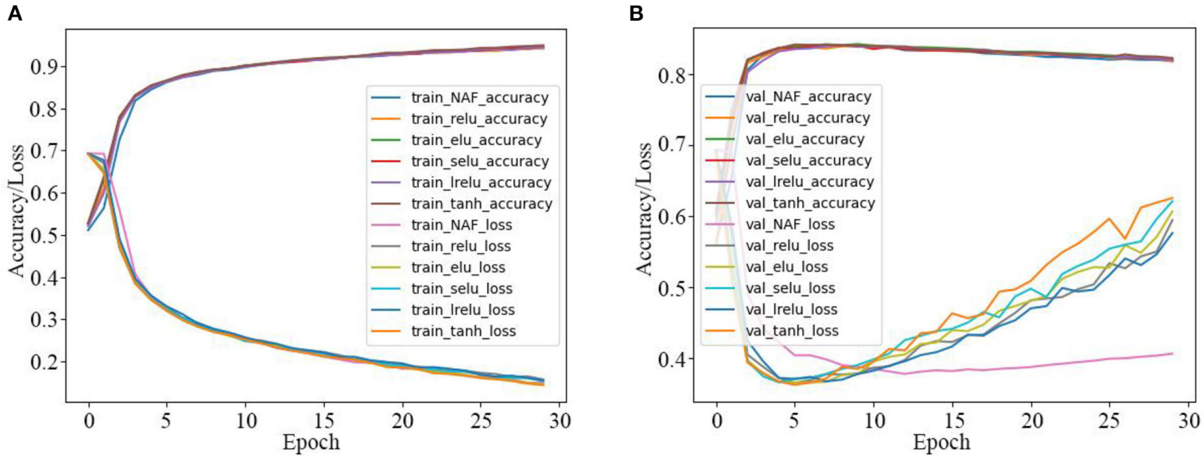

FIGURE 9
Training and test curves of GRU model for IMDB sentiment classification. **(A)** Training curves. **(B)** Tested sentiment classification curves.

ZNN model

As a special type of RNN, the ZNN model plays an important role in solving time-varying problems in the past few years. Following the methods in Zhang et al. (2002), the ZNN model for solving DSE problems can be constructed as below.

Firstly, according to Equation (5), a dynamic error function *E(t)* is adopted.

$$E(t) = A(t)X(t) - X(t)B(t) + C(t) \tag{6}$$

where $E(t) \in \mathbb{R}^{n\times n}$ is the error matrix, and its time derivative $dE(t)/dt$ should be negative-definite to ensure each element of *E(t)* converging to 0.

Secondly, the following formula is adopted for the convergence of *E(t)*.

$$\frac{dE(t)}{dt} = -\gamma\, \Psi(E(t)) \tag{7}$$

where $\gamma > 0$ is an adjustable parameter related to the convergence performance of the ZNN model. $\psi(\bullet) : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ is an nonlinear AF array, and ϕ *(•)* is the element of the matrix ψ*(•)*.

TABLE 5 GRU model with different Afs for IMDB sentiment classification.

No	AFs	Train accuracy (%)	Train loss	Val-accuracy (%)	Val-loss
1	ReLU	94.2500	0.1575	83.8900	0.5937
2	LReLU	94.4600	0.1553	84.0200	0.5755
3	ELU	94.7400	0.1475	84.1800	0.6061
4	SELU	94.5200	0.1533	84.0100	0.6202
5	Tanh	94.8000	0.1442	84.1300	0.6249
6	NAF	94.8100	0.1458	84.1900	0.4056

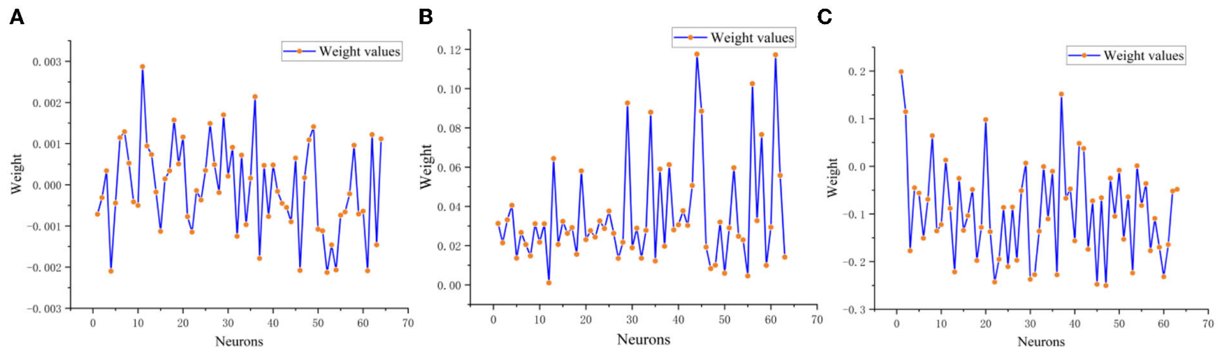

FIGURE 10
Output layer weights of SRNN, LSTM, and GRU model. **(A)** SRNN output layer weights. **(B)** LSTM output layer weights. **(C)** GRU output layer weights.

TABLE 6 Recently reported AFs for ZNN model (9).

No.	AFs	Expression
1	Linear activation function (LAF)	$\phi(x) = x$
2	Power-Sigmoid activation function (PSAF)	$\phi(x =) = \begin{cases} x^p, & \|x\| \geq 1 \\ \frac{1+e^{-\xi}-e^{-\xi x}}{1+e^{-\xi}+e^{-\xi x}}, & otherwise \end{cases}$
3	Sign-bi-power activation function (SBPAF)	$\phi(x) = \frac{1}{2} sgn^{\xi_1}(x) + \frac{1}{2} sgn^{\xi_2}(x)$
4	Versatile activation function (VAF)	$\phi(x) = \frac{b_1}{p}\exp(\|x\|^p)\|x\|^{1-p}\mathrm{sgn}x + b_2 x + b_3 \mathrm{sgn}(x)$

The time differential of equation (6) is

$$\frac{dE(t)}{dt} = \dot{A}(t)X(t) + A(t)\dot{X}(t) - \dot{X}(t)B(t) - X(t)\dot{B}(t) + \dot{C}(t) \tag{8}$$

At last, substituting equation (7) into (8), the ZNN model for solving DSE is realized.

$$A(t)\dot{X}(t) - \dot{X}(t)B(t) = -\dot{A}(t)X(t) + X(t)\dot{B}(t) - \dot{C}(t) - \gamma\,\Psi(E(t)) \tag{9}$$

ZNN model (9) is stable as long as AF ψ(•) is a monotonically increasing function (Zhang et al., 2007, 2008), and the recently reported AFs are listed in Table 6.

The NAF-based FTCRNN model

The FTCRNN model is obtained by introducing the NAF (1) to the ZNN model (9), and its expression is

$$\dot{E}(t) = -\gamma[\alpha_0(E(t)) + \alpha_1 sign(E(t))\left|E^{2\phi_1-1}(t)\right| + \alpha_2 sign(E(t))\left|E^{2\phi_2-1}(t)\right|] \tag{10}$$

Then, the FTCRNN model for solving DSE is shown as below.

$$A(t)\dot{X}(t) - \dot{X}(t)B(t) = -\dot{A}(t)X(t) + X(t)\dot{B}(t) - \dot{C}(t) + \dot{E}(t) \tag{11}$$

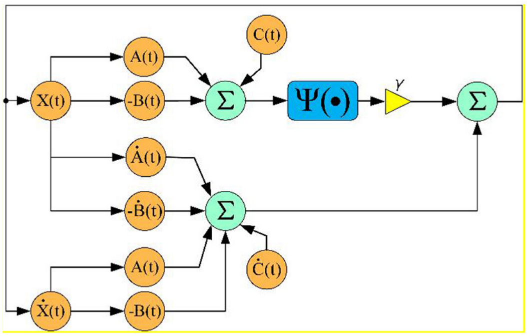

FIGURE 11
FTCRNN model algorithm diagram.

To further analyze its robustness, the FTCRNN model with additive noise *N(t)* is presented as follows.

$$A(t)\dot{X}(t) - \dot{X}(t)B(t) = -\dot{A}(t)X(t) + X(t)\dot{B}(t) - \dot{C}(t) + \dot{E}(t) + N(t) \qquad (12)$$

The corresponding algorithm diagram of the FTCRNN model for solving TODSE problem is presented in Figure 11. *A(t)*, *B(t)*, *C(t)*, *X(t)*, and $\psi(\bullet)$ are the previous defined matrices. Each orange circle represents a time-varying matrix or their derivative matrix, and the number of neurons in each orange circle depends on the number of matrix elements. Besides, the convergence factor γ is set as 1 for fair comparison.

FTCRNN analysis

In order to verify the fixed-time convergence and robustness to noise of the proposed FTCRNN model, the following Lemma 1 is presented in advance.

Lemma 1 (Chen et al., 2020). If $V(\bullet)$: $R^n \rightarrow R + \bigcup\{0\}$ is a continuous radially unbounded function, and the following two conditions hold.

(I) $V(e(t)) = 0 \Leftrightarrow e(t) = 0$;

(II) Any *e(t)* in system (13) satisfies

$$\dot{V}(e(t)) \le -\alpha V^{\xi}(e(t)) - \beta V^{\eta}(e(t)) - \delta V(e(t)), \quad t \in [0, +\infty] \qquad (13)$$

where α, β, $\delta > 0$, $0 < \xi < 1$, and $\eta > 1$. Then the dynamic system (13) is fixed-time stability, and

$$T_{\mathrm{m}} = \frac{1}{\delta(1-\xi)}\ln(1+\frac{\delta}{\alpha}) + \frac{1}{\delta(\eta-1)}\ln(1+\frac{\delta}{\beta}) \qquad (14)$$

Definition 1 (Chen et al., 2020). The dynamic system (13) achieves finite-time stability, if there exists a constant *T(e(0))* > 0 such that $\lim_{t\rightarrow T(e(0))} ||e(t)||_1$ and $\|e(t)\|_1 = 0$ for ∀ *t* > *T* *(e(0))*, where *T(e(0))* is the settling time.

Definition 2 (Chen et al., 2020). The dynamic system (13) achieves fixed-time stability, if two conditions are satisfied: (i) The dynamic system (13) achieves finite-time stability; (ii) For any *e(0)*, there exists a fixed constant $T_m > 0$ such that $T(e(0)) \le T_m$.

Proof: Let $U(s) = V^{1-\xi}(s)$, then we have

$$\dot{V}(s) = \frac{1}{1-\xi}\dot{U}(s)V^{\xi}(s) \qquad (15)$$

Combining Equations (13, 15) yields

$$\frac{1}{1-\xi}\dot{U}(e(t))V^{\xi}(e(t)) \le -\alpha V^{\xi}(e(t)) - \beta V^{\eta}(e(t)) - \delta V(e(t)) \qquad (16)$$

Then, we can rewrite Equation (16) as

$$\dot{U}(e(t)) \le -\alpha(1-\xi) \qquad (17)$$

Because $\dot{U}(e(t)) \le -\alpha(1-\xi)$, there always exists a constant $T(e(0)) = \frac{U(e(0))}{\alpha(1-\zeta)} > 0$, such that $\lim_{t\rightarrow T(e(0))} U(e(t)) = 0$ and $U(e(t)) = 0$ for $\forall t > T(e(0))$. In view of $U(e(0)) = 0 \Leftrightarrow V(e(t)) = 0 \Leftrightarrow e(t) = 0$, so there exists a constant $T(e(0)) > 0$ such that $\lim_{t\rightarrow T(e(0))} ||e(t)||_1 = 0$ and $||e(t)||_1 = 0$ for $\forall t > T(e(0))$. According to Definition 1, the dynamic system (13) is finite-time stable.

Equation (16) can be written in the following form

$$\frac{dt}{dV(e(t))} \ge \frac{1}{-\alpha V^{\xi}(e(t)) - \beta V^{\eta}(e(t)) - \delta V(e(t))} \qquad (18)$$

and we have

$$T(e(0)) \le \int_0^{V(e(0))} \frac{1}{\alpha s^{\xi} + \beta s^{\eta} + \delta s} ds \qquad (19)$$

Then, the analysis of Equation (19) should be divided into the following two cases.

(i) If $0 \le V(e(0)) \le 1$,

$$T(e(0)) \le \int_0^1 \frac{1}{\alpha s^{\xi} + \beta s^{\eta} + \delta s} ds \le \int_0^1 \frac{1}{\alpha s^{\xi} + \delta s} ds \qquad (20)$$

Let $\omega = s^{1-\xi}$, we have $d\omega = (1-\zeta)s^{-\xi}ds$, thus

$$\int_0^1 \frac{1}{\alpha s^{\xi} + \delta s} ds = \frac{1}{1-\xi}\int_0^1 \frac{s^{\xi}}{\alpha s^{\xi} + \delta s} d\omega = \frac{1}{\delta(1-\xi)}\ln(1+\frac{\delta}{\alpha}) \qquad (21)$$

Therefore, we can obtain

$$T(e(0)) \le \frac{1}{\delta(1-\xi)}\ln(1+\frac{\delta}{\alpha}) \qquad (22)$$

(ii) If $V(e(0)) \leq 1$,

$$\begin{aligned} T(e(0)) &\leq \int_0^1 \frac{1}{\alpha s^{\xi} + \beta s^{\eta} + \delta s} ds + \int_1^{V(e(0))} \frac{1}{\alpha s^{\xi} + \beta s^{\eta} + \delta s} ds \\ &\leq \int_0^1 \frac{1}{\alpha s^{\xi} + \delta s} ds + \int_1^{+\infty} \frac{1}{b s^{\eta} + \delta s} ds \end{aligned} \tag{23}$$

Let $c = s^{1-\eta}$, we have $dc = (1-\eta)s^{-\eta}ds$, thus

$$\begin{aligned} \int_1^{+\infty} \frac{1}{\beta s^{\eta} + \delta s} ds &= \frac{1}{1-\eta} \int_1^0 \frac{s^{\eta}}{\beta s^{\eta} + \delta s} dc \\ &= \frac{1}{\delta(\eta - 1)} \ln(1 + \frac{\delta}{\beta}) \end{aligned} \tag{24}$$

Therefore, we can obtain

$$T(e(0)) \leq \frac{1}{\delta(\eta - 1)} \ln(1 + \frac{\delta}{\beta}) \tag{25}$$

According to Definition 2, we can obtain that dynamic system (13) is stable in fixed-time T_m.

$$T_m \leq \frac{1}{\delta(1-\xi)} \ln(1 + \frac{\delta}{\alpha}) + \frac{1}{\delta(\eta - 1)} \ln(1 + \frac{\delta}{\beta}) \tag{26}$$

The proof of Lemma 1 is completed.

According to Lemma 1, we will verify the fixed-time convergence and robustness to noises of the proposed FTCRNN model in the following two cases.

Case 1: Convergence analysis without noise

Theorem 1. If the solution of DSE (1) exists, the state solution *X(t)* of FTCRNN model (11) will converge to the theoretical solution *X*(t)* of DSE (1) in fixed-time T_s.

$$T_s \leq \frac{1}{2\alpha_0(1-\phi_1)} \ln(1 + \frac{\alpha_0}{\alpha_1}) + \frac{1}{2\alpha_0(\phi_2 - 1)} \ln(1 + \frac{\alpha_0}{\alpha_2})$$

Proof. According to Equation (10), the *ij*th dynamic error function $e_{ij}(t)$ of FTCRNN model (11) can be expressed as

$$\begin{aligned} \dot{e}_{ij}(t) &= -\gamma \Big[\alpha_0(e_{ij}(t)) + \alpha_1 sign(e_{ij}(t)) \left|{e_{ij}}^{2\phi_1 - 1}(t)\right| \\ &+ \alpha_2 sign(e_{ij}(t)) \left|{e_{ij}}^{2\phi_2 - 1}(t)\right|\Big] \end{aligned} \tag{27}$$

When $\gamma = 1$, we choose the following Lyapunov function.

$$V(e_{ij}(t)) = \frac{1}{2}\left|e_{ij}(t)\right|^2 \tag{28}$$

$$\begin{aligned} \dot{V}(e_{ij}(t)) &= e_{ij}(t)\dot{e}_{ij}(t) = e_{ij}(t) \times \Big\{-\gamma \Big[\alpha_0(e_{ij}(t)) \\ &+ \alpha_1 sign(e_{ij}(t)) \left|{e_{ij}}^{2\phi_1 - 1}(t)\right| \\ &+ \alpha_2 sign(e_{ij}(t)) \left|{e_{ij}}^{2\phi_2 - 1}(t)\right|\Big]\Big\} \\ &= -\gamma \left(\alpha_0 e_{ij}^2(t) + \alpha_1 \left|{e_{ij}}^{2\phi_1}(t)\right| + \alpha_2 \left|{e_{ij}}^{2\phi_2}(t)\right|\right) \\ &= -\Big(2\alpha_0 V(e_{ij}(t)) + 2\alpha_1 V^{\phi_1}(e_{ij}(t)) \\ &+ 2\alpha_2 V^{\phi_2}(e_{ij}(t))\Big) \end{aligned} \tag{29}$$

According to Lemma 1, the convergent time of FTCRNN model (11) can be obtained.

$$T_{ij} \leq \frac{1}{2\alpha_0(1-\phi_1)} \ln(1 + \frac{\alpha_0}{\alpha_1}) + \frac{1}{2\alpha_0(\phi_2 - 1)} \ln(1 + \frac{\alpha_0}{\alpha_2}) \tag{30}$$

Case 2: Convergence analysis with noises

Noise interference is inevitable in the practical applications, therefore, it is necessary to consider the capability of anti-noises ability. Therefore, in this subsection, the robustness of the proposed FTCRNN model is analyzed, and the following theorem 2 guarantees the fixed-time stable of FTCRNN (12) in noisy environment.

Theorem 2. If FTCRNN model (12) is polluted by external noise *N(t)* with its *ij*th element satisfying $|n_{ij}(t)| \leq \gamma|e_{ij}(t)|$, where *σ ∈ (0, +∞)*, $|e_{ij}(t)|$ and $|n_{ij}(t)|$ are the absolute values of the *ij*th element of *E(t)* and *N(t)*, respectively. In addition, suppose $\gamma \times \min(\alpha, \alpha_1, \alpha_2) > \sigma$. Then, neural state solution *X(t)* of FTCRNN model (12) converges to the theoretical solution *X*(t)* of DSE (5) in fixed-time T_s.

$$\begin{aligned} T_s \leq & \frac{1}{(2\alpha_0 + \sigma)(1-\phi_1)} \ln(1 + \frac{2\alpha_0 + \sigma}{2\alpha_1}) \\ + & \frac{1}{(2\alpha_0 + \sigma)(\phi_2 - 1)} \ln(1 + \frac{2\alpha_0 + \sigma}{2\alpha_2}) \end{aligned} \tag{31}$$

where the parameters *a*, α_1, α_2, ϕ_1, ϕ_2 are defined similar as before.

Proof: The *ij*th dynamic error function $e_{ij}(t)$ of FTCRNN model (12) with noises can be expressed as

$$\begin{aligned} \dot{e}_{ij}(t) &= -\gamma \Big[\alpha_0(e_{ij}(t)) + \alpha_1 sign(e_{ij}(t)) \left|e_{ij}^{2\phi_1 - 1}(t)\right| \\ &+ \alpha_2 sign(e_{ij}(t)) \left|e_{ij}^{2\phi_2 - 1}(t)\right|\Big] - n_{ij}(t) \end{aligned} \tag{32}$$

When $\gamma = 1$, we choose the following Lyapunov function.

$$V(e_{ij}(t)) = \frac{1}{2}\left|e_{ij}(t)\right|^2 \tag{33}$$

$$\begin{aligned} &\dot{V}(e_{ij}(t)) = e_{ij}(t)\dot{e}_{ij}(t) = e_{ij}(t) \times \{-\gamma \big[A(e_{ij}(t)) \\ &+ \alpha_1 sign(e_{ij}(t)) \left|{e_{ij}}^{2\phi_1 - 1}(t)\right| + \alpha_2 sign(e_{ij}(t)) \left|{e_{ij}}^{2\phi_2 - 1}(t)\right|\big] \\ &-n_{ij}(t)\} = -\gamma \left(\alpha_0 e_{ij}^2(t) + \alpha_1 \left|{e_{ij}}^{2\phi_1}(t)\right| + \alpha_2 \left|{e_{ij}}^{2\phi_2}(t)\right|\right) \\ &-n_{ij}(t) \times e_{ij}(t) \leq -2\alpha_0 V(e_{ij}(t)) - 2\alpha_1 V^{\phi_1}(e_{i,j}(t)) \\ &-2\alpha_2 V^{\phi_2}(e_{ij}(t)) - \sigma V(e_{ij}(t)) = -2(\alpha_0 + \frac{1}{2}\sigma)V(e_{ij}(t)) \\ &-2\alpha_1 V^{\phi_1}(e_{ij}(t)) - 2\alpha_2 V^{\phi_2}(e_{ij}(t)) \end{aligned} \tag{34}$$

According to Lemma 1, the convergence time of FTCRNN model (12) can be obtained.

$$\begin{aligned} T_{ij} \leq & \frac{1}{(2\alpha_0 + \sigma)(1-\phi_1)} \ln(1 + \frac{2\alpha_0 + \sigma}{2\alpha_1}) \\ + & \frac{1}{(2\alpha_0 + \sigma)(\phi_2 - 1)} \ln(1 + \frac{2\alpha_0 + \sigma}{2\alpha_2}) \end{aligned} \tag{35}$$

Remark 1. As observed in Equations (30, 34), the parameters (α_0, α_1, α_2, ϕ_1, *and* ϕ_2) of NAF (1) are closely related to the convergent time of the FTCRNN model. Considering the above equations and the numerical simulation in the next section, we set $\alpha_0 = 50$, $\alpha_1 = \alpha_2 = 0.2$, $\phi_1 = 1.3$, $\phi_2 = 0.2$ to ensure the FTCRNN model achieve better convergence and robustness.

Dynamic problems simulation verification

In this subsection, three experiments using the NAF-based FTCRNN model are demonstrated. In example 1, a third-order DSE (TODSE) is solved by using the proposed NAF-based FTCRNN model. The FTCRNN-based robotic manipulator trajectory tracking application is applied in example 2. In example 3, an application of electric circuit currents computation using the NAF-based FTCRNN model is presented.

Example 1. TODSE problem solving

In this subsection, the proposed FTCRNN model is used to solve a TODSE problem on the basis of the following dynamic coefficient matrices. Additionally, the ZNN model (9) activated by other three AFs in Table 6 is also applied to solve the same TODSE problem for comparison.

$$A = \begin{pmatrix} \sin 3t & \cos 3t & \sin t \\ -\cos 3t & \sin 3t & \cos t \\ -\sin t & -\cos 2t & \sin 3t \end{pmatrix}$$

$$B = \begin{pmatrix} 20 & 0 & 10 \\ 0 & 30 & 15 \\ 0 & 15 & 20 \end{pmatrix}$$

$$C = \begin{pmatrix} \sin 3t & \cos 3t & \sin t \\ -\cos 3t & \sin 3t & \cos t \\ \sin t & \cos t & \sin 3t \end{pmatrix}$$

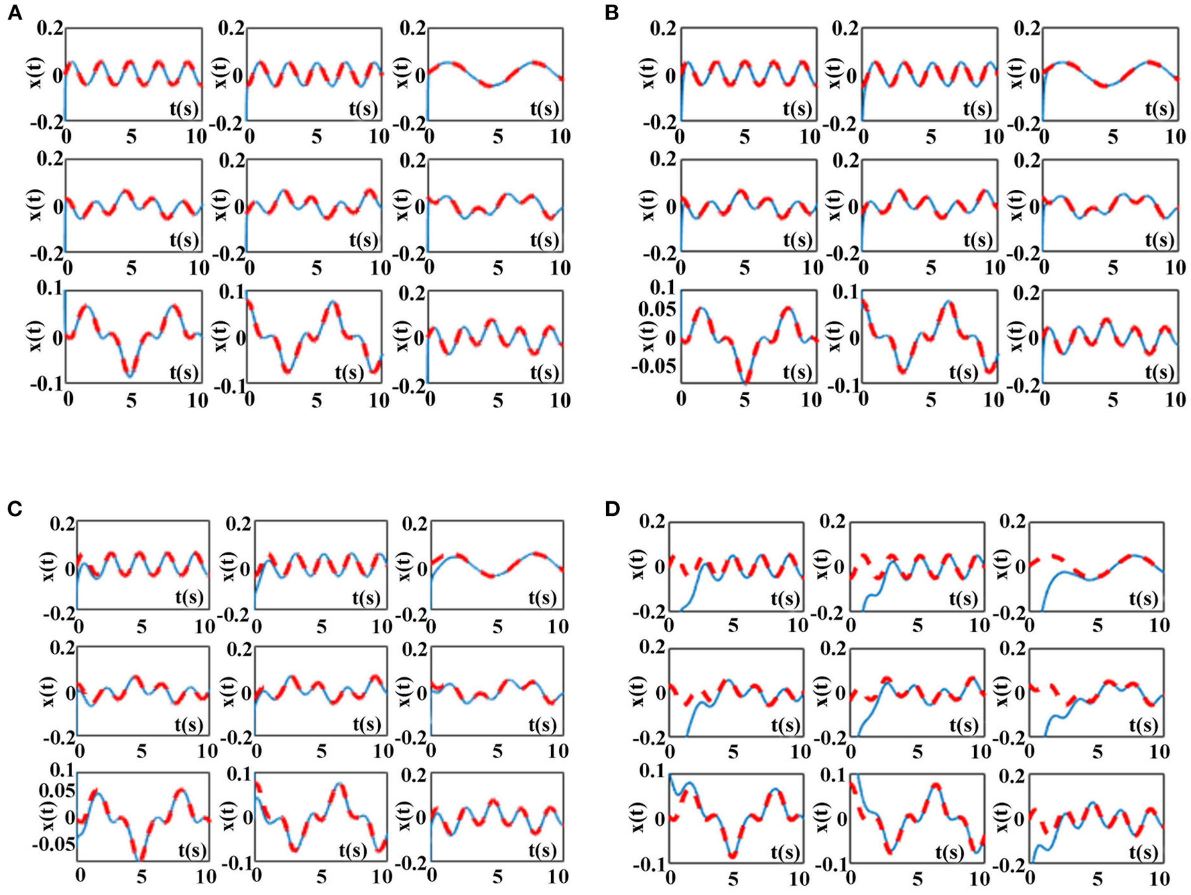

FIGURE 12
TODSE (5) solved by ZNN model (9) and the proposed NAF-based FTCRNN model (11) without noise. **(A)** Solved by FTCRNN model (11) without noise. **(B)** Solved by VAF-based ZNN model (9) without noise. **(C)** Solved by SBPAF-based ZNN model (9) without noise. **(D)** Solved by LAF-based ZNN model (9) without noise.

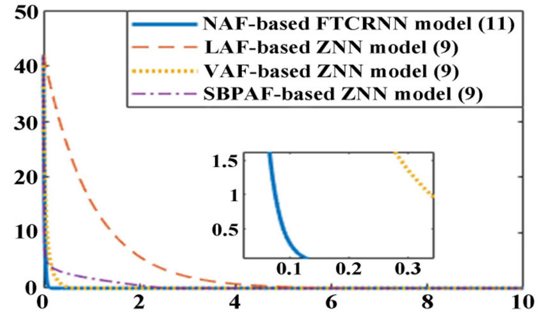

FIGURE 13
Residual errors of the ZNN model (9) and the NAF-based FTCRNN model (11) for solving TODSE without noise.

The simulation results of the ZNN model (9) activated by other three AFs in Table 6 and the proposed NAF-based FTCRNN model (11) for solving the above TODSE (5) in no noise environment are presented in Figure 12. The red dotted curves and blue solid curves are the theoretical solutions and neural state solutions of TODSE (5), respectively. As observe in Figure 12, the ZNN model (9) and the proposed NAF-based FTCRNN model all effectively solve the TODSE (5).

To clearly compare their effectiveness for solving TODSE (5) in no noise environment, the simulated residual errors $\|A(t)X(t) - X(t)B(t) + C(t)\|_F$ of the ZNN model (9) activated by other three AFs in Table 6 and the proposed NAF-based FTCRNN model are presented in Figure 13. It is obvious that all the models possess the ability to converge to the theoretical solution of TODSE (5), but their effectiveness are quite different. The proposed FTCRNN model (11) spends only about 0.15 s to find the theoretical solution of TODSE (5), which is the most effective candidate for solving TODSE in no noise environment.

Moreover, according to Equation (30), the convergent time of the FTCRNN model in no noise environment can be calculated.

$$T_s \le \frac{1}{2\alpha_0(1-\phi_1)}\ln(1+\frac{\alpha_0}{\alpha_1}) + \frac{1}{2\alpha_0(\phi_2-1)}\ln(1+\frac{\alpha_0}{\alpha_2}) \approx 0.7s \qquad (36)$$

The theoretical convergent time of the FTCRNN model in no noise environment is 0.7 s, and the simulated convergent time is about 0.15 s, which prove that the theoretical analysis result is consistent with the experimental result.

Considering the inevitable noises in practical application, ZNN model (9) activated by other three AFs in Table 6 and the proposed NAF-based FTCRNN model (12) are also applied to solve the TODSE (5) in noisy environment, and the dynamic mixed-noise matrix *N(t)* is presented as below. The corresponding simulation results for solving TODSE (5) in noise-polluted environment are presented in Figures 14, 15.

$$N(t) = \begin{pmatrix} 1.5\sin t & 0.5t & 1.5 \\ 1.5e^{-t} & 0.5+0.1t & 1.5\cos t + 0.1 \\ 1.5\sin t + 0.1t & 1.5\cos t + e^{-t} & e^{-t}+0.1 \end{pmatrix} \qquad (37)$$

As shown in Figure 14, the proposed NAF-based FTCRNN model (12) and the VAF-based ZNN model (9) still effectively solve TODSE (5) in noisy environment, but the ZNN model (9) activated by the LAF and SBPAF fail to solve TODSE (5) due to the influence of noises.

The residual errors of the ZNN model (9) activated by other three AFs in Table 6 and the proposed FTCRNN model (12) are also presented in Figure 15 for further comparison. However, it can be seen from Figure 15 that the convergence performance of the FTCRNN model and the VAF-based ZNN model is completely different. With the increase of solution time, the residual error of the VAF-based ZNN model is increasing, while the residual error generated by the FTCRNN model remains stable at zero.

Moreover, according to Equation (34), the theoretical convergent time of the FTCRNN model in noisy environment can be calculated.

$$T_s \le \frac{1}{(2\alpha_0+\sigma)(1-\phi_1)}\ln(1+\frac{2\alpha_0+\sigma}{2\alpha_1}) + \frac{1}{(2\alpha_0+\sigma)(\phi_2-1)}\ln(1+\frac{2\alpha_0+\sigma}{2\alpha_2}) \approx 0.7s \qquad (38)$$

The theoretical convergent time of the FTCRNN model in noisy environment is 0.7 s, and the simulated convergent time is about 0.2 s, which further indicates the theoretical analysis result is consistent with the experimental result.

For showing the different performances of the four AFs based models, comprehensive comparisons are listed in Table 7. Obviously, the NAF-based FTCRNN model is the best one not only in convergence performance but also in robustness. Besides, only the NAF-based FTCRNN model realizes fast convergence under the condition of noise interference.

Based on the above analysis, we can conclude that the proposed FTCRNN model has better robustness and effectiveness than recently reported works.

Remark 2. According to the hypothesis in Theorem 2 and the noise matrix *N(t)*, we have $|n_{ij}(t)| \le \sigma|e_{ij}(t)|$ and $|n_{ij}(t)| \le 2.5$. We can also have $|e_{ij}(t)| \approx 50$ from Figure 15. Then, the value of parameter σ is $\sigma \ge 0.005$, and we set $\sigma = 0.07$.

Example 2. Application to robotic manipulator

The research on robots becomes popular in recent years (Jin et al., 2020; Zhang et al., 2020a, 2021; Jin, 2021b). Therefore, the FTCRNN-based RM trajectory tracking application is realized to further verify the practical application feasibility of the FTCRNN model.

The three-dimensional structure and geometrical model of a RM are presented in Figure 16. As observed in Figure 16A, the RM consists of a movable platform with a manipulator with six joints, and its geometrical model is depicted in Figure 16B.

Based on the modeling method in Xiao and Zhang (2014a) and Zhao et al. (2021), its kinematic matrix equation in velocity level is presented as follows.

$$\dot{r}_m = B(\phi,\theta)\dot{\Theta} \tag{39}$$

where r_m is an end-effector position vector, $\Theta = [\varphi^T, \theta^T]^T$ denotes a combined RM angle vector. Besides, $\dot{r}_m \in \mathbb{R}^m$ and $\dot{\Theta}$ are time derivatives of r_m and Θ, respectively. $B(\phi,\theta) \in \mathbb{R}^{m\times(2+n)}$ is defined as

$$B(\phi,\theta) = \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} + J(\phi,\theta)\begin{bmatrix} N & 0 \\ 0 & I \end{bmatrix} \tag{40}$$

where

$$M = \frac{r}{2}\begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}\begin{bmatrix} 1 & 1 \\ -d/b & d/b \end{bmatrix}, N = \frac{r}{2\mathrm{b}}\begin{bmatrix} -1 \\ 1 \end{bmatrix}^T \tag{41}$$

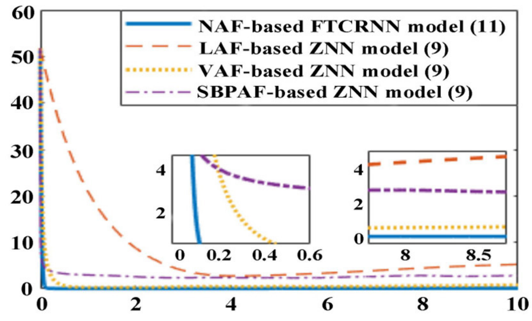

FIGURE 15
Residual errors of ZNN model (9) and NAF-based FTCRNN model for solving TODSE in noisy environment.

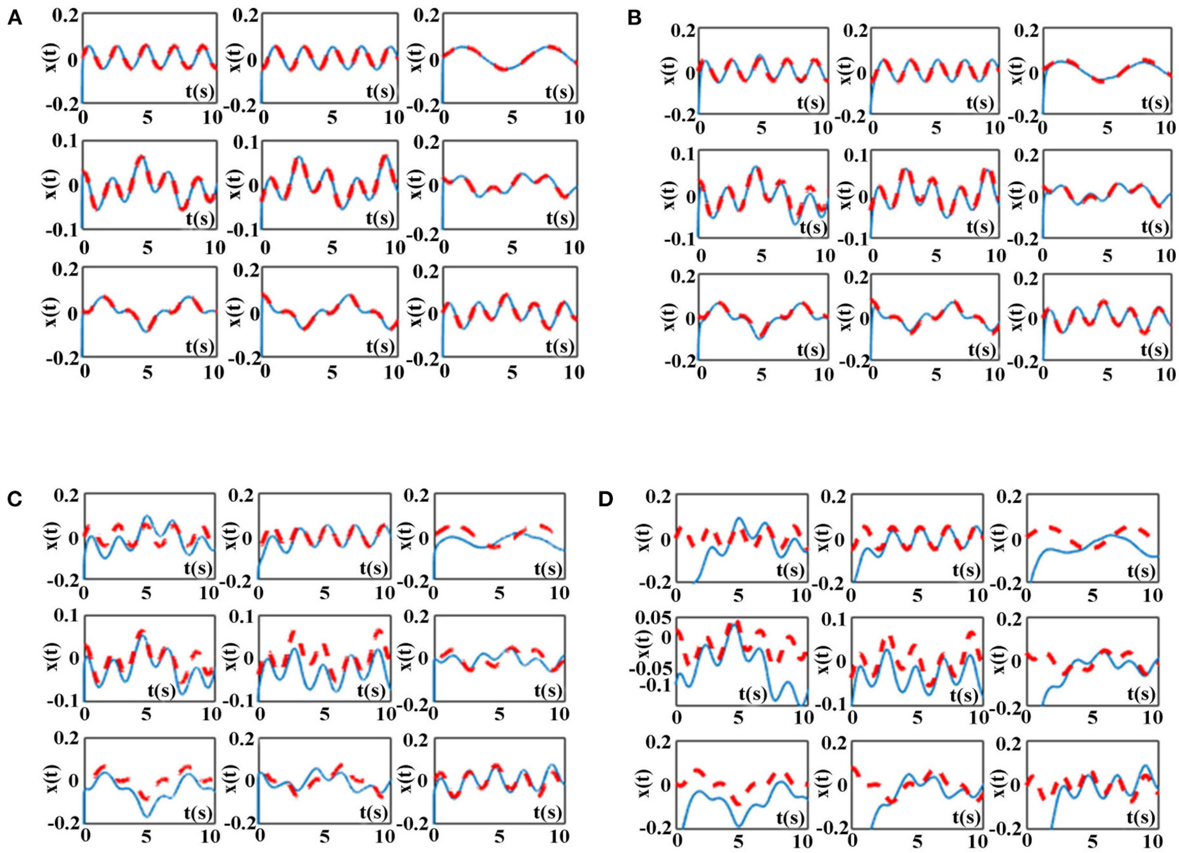

FIGURE 14
TODSE (5) solved by ZNN model (9) and the proposed NAF-based FTCRNN model (12) in noisy environment. **(A)** Solved by FTCRNN model (12) in noisy environment **(B)** Solved by VAF-based ZNN model (9) in noisy environment. **(C)** Solved by SBPAF-based ZNN model (9) in noisy environment **(D)** Solved by LAF-based ZNN model (9) in noisy environment.

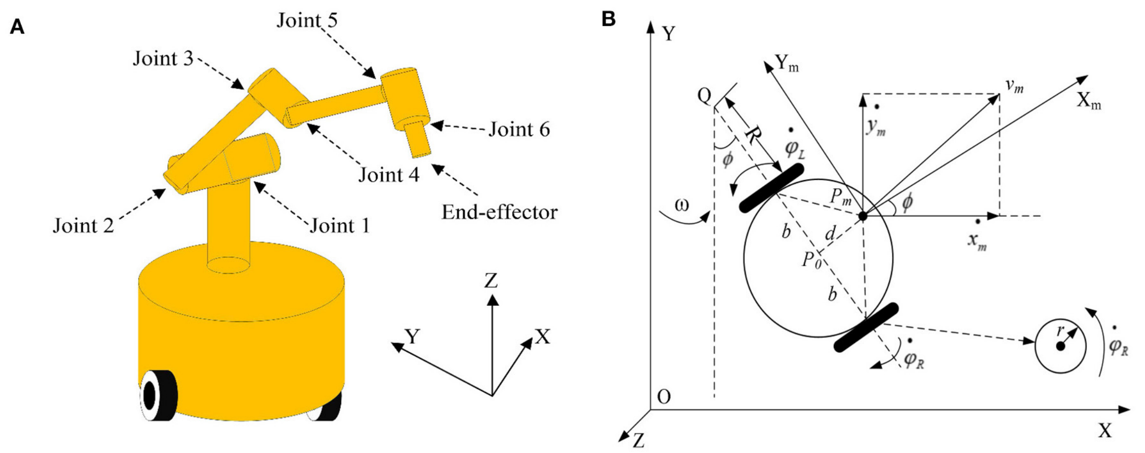

FIGURE 16
Modeling of the RM. **(A)** Three-dimensional structure of the RM. **(B)** Geometrical model of the RM.

TABLE 7 Comparisons of the LAF, SBPAF, VAF, and NAF based models for solving TODSE.

	LAF-based ZNN (9)	SBPAF-based ZNN (9)	VAF-based ZNN (9)	FTCRNN
Convergence	Poor	Poor	Better	Best
Robustness	Weak	Weak	Strong	Very strong
Convergence time in ideal environment	5.5 s	2.5 s	0.6 s	0.15 s
Convergence time in noise environment	/	/	/	0.2 s

and $J(\varphi, \theta)$ is the Jacobian matrix.

Here, $r_{md}(t) \in \mathbb{R}^m$ is the desired path to be tracked.$r_m(t)$ is the actual path of the RM end-effector, and we have

$$f(\Theta, t) = r_m(t) \to r_{md}(t) \tag{42}$$

where $f(\bullet)$ stands for a continuous non-linear kinematic map of the RM.

Then, differentiating Equation (37) yields the following velocity level kinematic equation.

$$B(\phi, \theta)\dot{\Theta} = \dot{r}_m(t) \to \dot{r}_{md}(t) \tag{43}$$

On the basis of the above discussion, it is clear that the desired path $r_{md}(t) \in \mathbb{R}^m$ is given in advance, and the joint and wheel trajectories of the RM are required to be calculated by the RNN models. Therefore, the RNN models for solving the above tracking task are constructed as the following steps.

Firstly, a vector-valued error matrix is defined.

$$E(t) = r_{md}(t) - r_m(t) \tag{44}$$

The trajectory tracking task is transformed to ensure each element $e_i(t)$ of $E(t)$ converging to 0, and the following formula is used to guarantee its convergence.

$$\dot{E}(t) = -\lambda\Psi(E(t)) \tag{45}$$

For better comparison, both of the FTCRNN model (12) and the SBPAF-based ZNN model (9) are applied to ensure the convergence of *E(t)* in noisy environment, and the corresponding models are shown as follows.

$$B\dot{\Theta} = \dot{r}_{md}(t) + \lambda\Psi_1\left(r_{md}(t) - r_m(t)\right) + N(t) \tag{46}$$

$$B\dot{\Theta} = \dot{r}_{md}(t) + \lambda\Psi_2\left(r_{md}(t) - r_m(t)\right) + N(t) \tag{47}$$

where $\psi_1(\bullet)$ is the proposed NAF in Equation (1), and $\psi_2(\bullet)$ is the SBPAF in Table 6.

The desired tracking path is a helical line, and the initial state of the robotic manipulator is set as $\Theta(0) = [\pi/6, \pi/3, \pi/6, \pi/3, \pi/3, \pi/3]^T$, and $N(t) = 0.025\text{cos}t$. The trajectory tracking experiment results of robotic manipulator synthesized by the proposed FTCRNN-based model (41) and the SBPAF-based ZNN model (42) are displayed in Figures 17, 18.

It can be seen from Figures 17, 18 that the FTCRNN-based model (41) successfully completes the tracking task in noisy environment, and the end-effector trajectory coincides with the desired path. Besides, the tracking errors of the FTCRNN-based model (41) are all <0.1 cm. However, the SBPAF-based ZNN model (42) fails to complete the tracking task due to the disturbance of additive noises, and tracking errors of the SBPAF-based ZNN model (42) are all more than 1.5 cm. Furthermore,

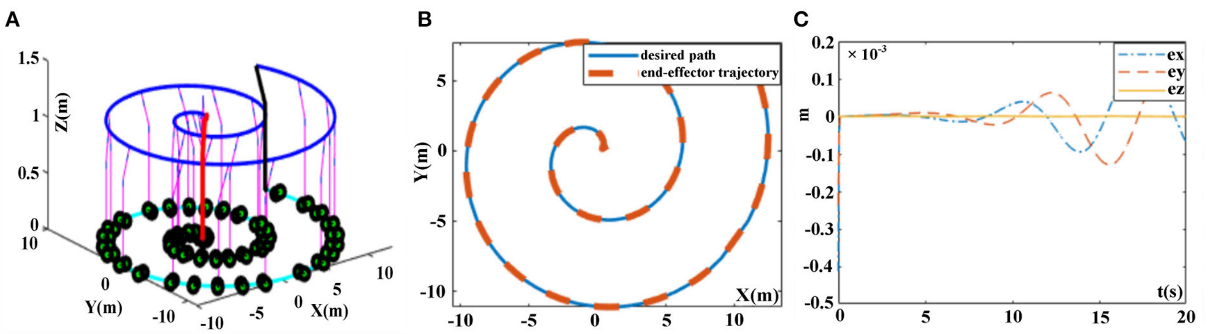

FIGURE 17
Trajectory tracking results synthesized by the FTCRNN-based model (41) in noisy environment. **(A)** Whole tracking motion trajectories. **(B)** Desired path and actual trajectory. **(C)** Tracking errors.

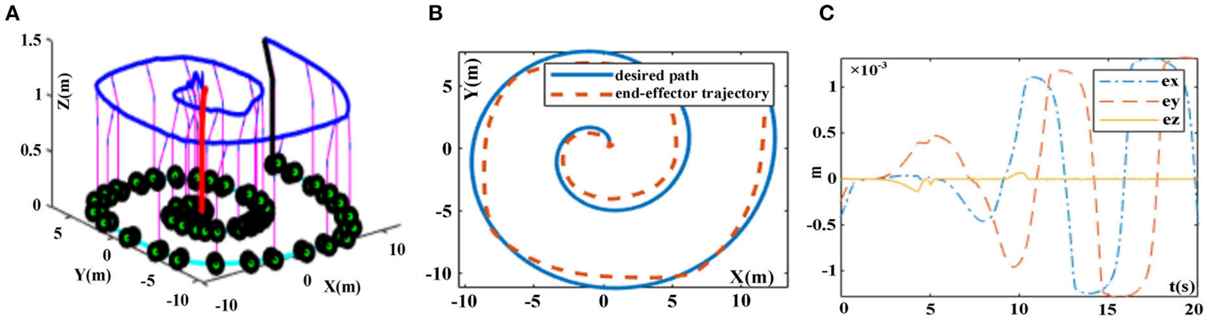

FIGURE 18
Trajectory tracking results synthesized by the ZNN-based model (42) in noisy environment. **(A)** Whole tracking motion trajectories. **(B)** Desired path and actual trajectory. **(C)** Tracking errors.

compared with the RM trajectory tracking methods in Xiao and Zhang (2014a) and Zhao et al. (2021), the most significant improvement of this work lies in its strong robustness to noises, which ensures the success of actual RM trajectory tracking application considering noises in reality.

Conclusion

In this paper, a NAF is proposed and employed in different RNN models to handle sentiment classification problem and dynamic problems solving. Experiment results of IMDB sentiment classification demonstrate that the NAF-based RNN models have better training and test accuracy, lower and more stable loss values than other AFs in Table 1 based RNN models. Besides, based on the NAF, the FTCRNN model is developed. The effectiveness and robustness of the proposed FTCRNN model are analyzed in theory and simulation experiment. Moreover, the FTCRNN-based robotic manipulator trajectory tracking application is carried out to verify the practical application feasibility of the FTCRNN model. Furthermore, the application of electric circuit currents computation further verifies the wide applicability of the FTCRNN model. This work provides a promising choice for the selection of AF of RNN models in sentiment classification and time-varying problems solving applications. The fixed-time convergence and strong robustness of the proposed NAF-based FTCRNN model guarantee its real-time online computing capability in noisy environment. Note that, only sentiment classification and time-varying problems solving of the NAF-based RNN models are considered in this paper. Thus, extending the applications of the NAF in different RNN models for text classification and translation needs future research.

Data availability statement

The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Author contributions

All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

Funding

This work was supported by the Guiding Plan Project of Scientific and Technological Innovation and Development in Changde (Grant/Award Number: 2020ZD37) and the Scientific Research Project from Hunan University of Arts and Science (Grant/Award Number: 21ZD03).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abdi, A., Shamsuddin, S. M., Hasan, S., and Piran, J. (2018). Automatic sentiment-oriented summarization of multi-documents using soft computing. *Soft Comput.* 4, 1–18. doi: 10.1007/s00500-018-3653-4

Azadeh, A., Ravanbakhsh, M., Rezaei-Malek, M., Sheikhalishahi, M., and Taheri-Moghaddam, A. (2017). Unique NSGA-II and MOPSO algorithms for improved dynamic cellular manufacturing systems considering human factors. *Appl. Math. Model.* 48, 655–672. doi: 10.1016/j.apm.2017.02.026

Chen, C., Li, L., Peng, H., Yang, Y., and Zhao, H. (2020). A new fixed-time stability theorem and its application to the fixed-time synchronization of neural networks. *Neural Netw.* 123, 412–419. doi: 10.1016/j.neunet.2019.12.028

Chen, X., and Xie, H. (2020). A structural topic modeling-based bibliometric study of sentiment analysis literature. *Cogn. Comput.* 12, 1097–1129. doi: 10.1007/s12559-020-09745-1

Darouach, M. (2006). Solution to Sylvester equation associated to linear descriptor systems. *Syst. Control Lett.* 55, 835–838. doi: 10.1016/j.sysconle.2006.04.004

Gong, J., and Jin, J. (2021). A better robustness and fast convergence zeroing neural network for solving dynamic nonlinear equations. *Neural Comput. Applic.* doi: 10.1007/s00521-020-05617-9

He, B., Zhang, Y., Zhou, Z., Wang, B., Liang, Y., Lang, J., et al. (2020). A neural network framework for predicting the tissue-of-origin of 15 common cancer types based on RNA-Seq data. *Front. Bioeng. Biotechnol.* 8, 737. doi: 10.3389/fbioe.2020.00737

Jin, J. (2021a). A robust zeroing neural network for solving dynamic nonlinear equations and its application to kinematic control of mobile manipulator. *Compl. Intell. Syst.* 7, 87–99. doi: 10.1007/s40747-020-00178-9

Jin, J. (2021b). An improved finite time convergence recurrent neural network with application to time-varying linear complex matrix equation solution. *Neural Proc. Lett.* 53, 777–786. doi: 10.1007/s11063-021-10426-9

Jin, J., and Gong, J. (2021). An interference-tolerant fast convergence zeroing neural network for dynamic matrix inversion and its application to mobile manipulator path tracking. *Alexand. Eng. J.* 60, 659–669. doi: 10.1016/j.aej.2020.09.059

Jin, J., and Qiu, L. (2022). A robust fast convergence zeroing neural network and its applications to dynamic Sylvester equation solving and robot trajectory tracking. *J. Franklin Inst.* 359, 3183–3209. doi: 10.1016/j.jfranklin.2022.02.022

Jin, J., Zhao, L., Li, M., Yu, F., and Xi, Z. (2020). Improved zeroing neural networks for finite time solving nonlinear equations. *Neural Comput. Applic.* 32, 4151–4160. doi: 10.1007/s00521-019-04622-x

Jin, J., Zhu, J., Gong, J., and Chen, W. (2022a). Novel activation functions-based ZNN models for fixed-time solving dynamic sylvester equation. *Neural Comput. Applic.* 34, 14297–14315. doi: 10.1007/s00521-022-06905-2

Jin, J., Zhu, J., Zhao, L., Chen, L., and Gong, J. (2022b). A robust predefined-time convergence zeroing neural network for dynamic matrix inversion. *IEEE Trans. Cybern.* doi: 10.1109/TCYB.2022.3179312. [Epub ahead of print].

Jin, L., Li, J., Sun, Z., Lu, J., and Wang, F. (2022c). Neural dynamics for computing perturbed nonlinear equations applied to ACP-based lower limb motion intention recognition. *IEEE Trans. Syst. Man Cyber. Syst.* 52, 5105–5113. doi: 10.1109/TSMC.2021.3114213

Jin, L., Wei, L., and Li, S. (2022d). Gradient-based differential neural-solution to time-dependent nonlinear optimization. *IEEE Trans. Automat. Contr.* doi: 10.1109/TAC.2022.3144135. [Epub ahead of print].

Li, S., Chen, S., and Liu, B. (2013). Accelerating a recurrent neural network to finite-time convergence for solving time-varying Sylvester equation by using a sign-bi-power activation function. *Inform. Proc. Lett.* 37, 189–205. doi: 10.1007/s11063-012-9241-1

Liu, M., Chen, L., Du, X., Jin, L., and Shang, M. (2021). Activated gradients for deep neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* doi: 10.1109/TNNLS.2021.3106044. [Epub ahead of print].

Liu, M., Zhang, X., Shang, M., and Jin, L. (2022). Gradient-based differential kWTA network with application to competitive coordination of multiple robots. *IEEE/CAA J. Autom. Sin.* 9, 1452–1463. doi: 10.1109/JAS.2022.105731

Saeed, R. M., Rady, S., and Gharib, T. F. (2021). Optimizing sentiment classification for arabic opinion texts. *Cogn. Comput.* 13, 164–178. doi: 10.1007/s12559-020-09771-z

Wan, Q., Yan, Z., Li, F., Chen, S., and Liu, J. (2022a). Complex dynamics in a Hopfield neural network under electromagnetic induction and electromagnetic radiation. *Chaos* 32, 073107. doi: 10.1063/5.0095384

Wan, Q., Yan, Z., Li, F., Liu, J., and Chen, S. (2022b). Multistable dynamics in a hopfield neural network under electromagnetic radiation and dual bias currents. *Nonlinear Dyn.* 109, 2085–2101. doi: 10.1007/s11071-022-07544-x

Wang, Z., and Lin, Z. (2020). Optimal feature selection for learning-based algorithms for sentiment classification. *Cogn. Comput.* 12, 238–248. doi: 10.1007/s12559-019-09669-5

Xiao, L., and He, Y. (2021). A noise-suppression ZNN model with new variable parameter for dynamic sylvester equation. *IEEE Trans. Industr. Inf.* 17, 7513–7522. doi: 10.1109/TII.2021.3058343

Xiao, L., Liao, B., Li, S., and Chen, K. (2018). Nonlinear recurrent neural networks for finite-time solution of general time-varying linear matrix equations. *Neural Netw.* 98, 102–113. doi: 10.1016/j.neunet.2017.11.011

Xiao, L., and Zhang, Y. (2014a). A new performance index for the repetitive motion of mobile manipulators. *IEEE Trans. Cybern.* 44, 280–292. doi: 10.1109/TCYB.2013.2253461

Xiao, L., and Zhang, Y. (2014b). From different Zhang functions to various ZNN models accelerated to finite-time convergence for time-varying linear matrix equation. *Neural Process. Lett.* 39, 309–326. doi: 10.1007/s11063-013-9306-9

Xiao, L., Zhang, Z., and Li, S. (2019). Solving time-varying system of nonlinear equations by finite-time recurrent neural networks with application to motion tracking of robot manipulators. *IEEE Trans. Syst. Man Cyber. Syst.* 49, 2210–2220. doi: 10.1109/TSMC.2018.2836968

Yaslan, Y., and Aldo, D. (2017). A comparison study on active learning integrated ensemble approaches in sentiment analysis. *Comput. Electr. Eng.* 57, 311–323. doi: 10.1016/j.compeleceng.2016.11.015

Yu, F., Chen, H., Kong, X., Yu, Q., Cai, S., Huang, Y., et al. (2022a). Dynamic analysis and application in medical digital image watermarking of a new multi-scroll neural network with quartic nonlinear memristor. *Euro. Phys. J. Plus* 137, 434. doi: 10.1140/epjp/s13360-022-02652-4

Yu, F., Liu, L., Xiao, L., Li, K., S., and Cai. (2019). A robust and fixed-time zeroing neural dynamics for computing time-variant nonlinear equation using a novel nonlinear activation function. *Neurocomputing* 350, 108–116. doi: 10.1016/j.neucom.2019.03.053

Yu, F., Yu, Q., Chen, H., Kong, X., Mokbel, A. A. M., Cai, S., et al. (2022b). Dynamic analysis and audio encryption application in IoT of a multi-scroll fractional-order memristive hopfield neural network. *Fractal Fract.* 6, 370. doi: 10.3390/fractalfract6070370

Zablith, F., and Osman, I. H. (2019). Review modus: text classification and sentiment prediction of unstructured reviews using a hybrid combination of machine learning and evaluation models. *Appl. Math. Model.* 71, 569–583. doi: 10.1016/j.apm.2019.02.032

Zhang, Y., Chen, K., Li, X., Yi, C., and Zhu, H. (2008). "Simulink modeling and comparison of Zhang neural networks and gradient neural networks for time-varying Lyapunov equation solving," in *Proceedings of IEEE International Conference on Natural Computation* (Jinan), 521–525. doi: 10.1109/ICNC.2008.47

Zhang, Y., Jiang, D., and Wang, J. (2002). A recurrent neural network for solving sylvester equation with time-varying coefficients. *IEEE Trans. Neural Netw.* 13, 1053–1063. doi: 10.1109/TNN.2002.1031938

Zhang, Y., Peng, H., and Zhang, F. (2007). "Neural network for linear time-varying equation solving and its robotic application," in *2007 International Conference on Machine Learning and Cybernetics* (Hong Kong), 3543–3548. doi: 10.1109/ICMLC.2007.4370761

Zhang, Z., Chen, S., Xie, J., and Yang, S. (2021). Two hybrid multiobjective motion planning schemes synthesized by recurrent neural networks for wheeled mobile robot manipulators. *IEEE Trans. Syst. Man Cybern. Syst.* 51, 3270–3281. doi: 10.1109/TSMC.2019.2920778

Zhang, Z., Chen, S., Zhu, X., and Yan, Z. (2020a). Two hybrid end-effector posture-maintaining and obstacle-limits avoidance schemes for redundant robot manipulators. *IEEE Trans. Indust. Inform.* 16, 754–763. doi: 10.1109/TII.2019.2922694

Zhang, Z., Chen, T., Wang, M., and Zheng, L. (2020b). An exponential-type anti-noise varying-gain network for solving disturbed time-varying inversion systems. *IEEE Trans. Neural Netw. Learn. Syst.* 31, 3414–3427. doi: 10.1109/TNNLS.2019.2944485

Zhang, Z., Fu, T., Yan, Z., Jin, L., Xiao, L., Sun, Y., et al. (2018a). A varying-parameter convergent-differential neural network for solving joint-angular-drift problems of redundant robot manipulators. *IEEE/ASME Trans. Mechatron.* 23, 679–689. doi: 10.1109/TMECH.2018.2799724

Zhang, Z., Kong, L., Zheng, L., Zhang, P., Qu, X., Liao, B., et al. (2020c). Robustness analysis of a power-type varying-parameter recurrent neural network for solving time-varying QM and QP problems and applications. *IEEE Trans. Syst. Man Cybern. Syst.* 50, 5106–5118. doi: 10.1109/TSMC.2018.2866843

Zhang, Z., Lu, Y., Zheng, L., Li, S., Yu, Z., and Li, Y. (2018b). A new varying-parameter convergent-differential neural-network for solving time-varying convex QP problem constrained by linear-equality. *IEEE Trans. Automat. Control* 63, 4110–4125. doi: 10.1109/TAC.2018.2810039

Zhang, Z., Zheng, L., Qiu, T., and Deng, F. (2020d). Varying-Parameter convergent-differential neural solution to time-varying overdetermined system of linear equations. *IEEE Trans. Automat. Control* 65, 874–881. doi: 10.1109/TAC.2019.2921681

Zhang, Z., Zheng, L., Weng, J., Mao, Y., Lu, W., and Xiao, L. (2018c). A new varying-parameter recurrent neural-network for online solution of time-varying sylvester equation. *IEEE Trans. Cybern.* 48, 3135–3148. doi: 10.1109/TCYB.2017.2760883

Zhao, L., Jin, J., and Gong, J. (2021). Robust zeroing neural network for fixed-time kinematic control of wheeled mobile robot in noise-polluted environment. *Math. Comp. Simulat.* 185, 289–307. doi: 10.1016/j.matcom.2020.12.030

Zhu, J., Jin, J., Chen, W., and Gong, J. (2022). A combined power activation function based convergent factor-variable ZNN model for solving dynamic matrix inversion. *Math. Comput. Simul.* 197, 291–307. doi: 10.1016/j.matcom.2022.02.019