IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 17, 2020, accepted August 31, 2020, date of publication September 3, 2020, date of current version September 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3021527

Evolving CNN-LSTM Models for Time Series Prediction Using Enhanced Grey Wolf Optimizer

HAILUN XIE[1], LI ZHANG[1], (Senior Member, IEEE), AND CHEE PENG LIM[2]

[1]Computational Intelligence Research Group, Department of Computer and Information Sciences, Faculty of Engineering and Environment, Northumbria University, Newcastle upon Tyne NE1 8ST, U.K.
[2]Institute for Intelligent Systems Research and Innovation, Deakin University, Melbourne, VIC 3216, Australia

Corresponding author: Li Zhang (li.zhang@northumbria.ac.uk)

This work was supported by Innovate UK Knowledge Transfer Partnership and Northumbria University under Global Challenges Research Fund.

ABSTRACT In this research, we propose an enhanced Grey Wolf Optimizer (GWO) for designing the evolving Convolutional Neural Network-Long Short-Term Memory (CNN-LSTM) networks for time series analysis. To overcome the probability of stagnation at local optima and a slow convergence rate of the classical GWO algorithm, the newly proposed variant incorporates four distinctive search mechanisms. They comprise a nonlinear exploration scheme for dynamic search territory adjustment, a chaotic leadership dispatching strategy among the dominant wolves, a rectified spiral local exploitation action, as well as probability distribution-based leader enhancement. The evolving CNN-LSTM models are subsequently devised using the proposed GWO variant, where the network topology and learning hyperparameters are optimized for time series prediction and classification tasks. Evaluated using a number of benchmark problems, the proposed GWO-optimized CNN-LSTM models produce statistically significant results over those from several classical search methods and advanced GWO and Particle Swarm Optimization variants. Comparing with the baseline methods, the CNN-LSTM networks devised by the proposed GWO variant offer better representational capacities to not only capture the vital feature interactions, but also encapsulate the sophisticated dependencies in complex temporal contexts for undertaking time-series tasks.

INDEX TERMS Evolutionary computation, Grey Wolf optimizer, time series prediction, and deep neural network.

I. INTRODUCTION

A time series is a sequence of data measured chronologically at a uniform time interval [1]. Time series measurements are prevalent in various domains, such as weather forecast [2], financial market prediction [3], physiological assessment [4] and video analysis [5]. Over the last several decades, many efforts have been made to develop effective time series forecasting models, which can be broadly classified into three categories: 1) statistical models, e.g. auto-regressive moving average (ARMA) [6] and auto-regressive integrated moving average (ARIMA) [7]; 2) machine learning models, e.g. Support Vector Regression (SVR) [8] and Artificial Neural Networks (ANN) [9]; 3) deep learning models, e.g. Recurrent Neural Networks (RNN) [10] and Long Short-Term Memory (LSTM) [11]. In particular, the LSTM network is regarded as the state-of-the-art time series forecasting model owing to its capability of learning long-term temporal dependencies through the design of gated units integrating activations of sigmoid and hyperbolic tangent functions.

Despite the progress achieved by LSTM, multi-variate time series forecasting remains a challenging task owing to the complex factors embedded in real-life sequential data, such as sophisticated dependencies, irregularity, randomness, cross-correlation among variables, as well as noise [12], [13]. Besides that, hyperparameters in relation to the configuration of LSTM, e.g. the number of hidden nodes, as well as the learning properties during the training process, e.g. learning rate, play vital roles in affecting the performance of the LSTM networks [14], [15]. In this regard, the identification of the optimal hyperparameter settings for LSTM networks remains a challenging task owing to the complexity of the problems at hand and the requirement of profound domain knowledge. The traditional manual trial-and-error fine-tuning process is likely to result in sub-optimal model representational capacities and ill-performed learning parameters, therefore compromising the performance of LSTM networks. In order to resolve the aforementioned challenges in dealing with time

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

series data as well as optimal learning configuration identification of LSTM networks, we incorporate two automatic processes into the vanilla LSTM structure, i.e. automatic feature extraction and optimal network configuration identification, to enhance the performance of the monotonous LSTM networks in tackling time series prediction. Essentially, Convolutional Neural Networks (CNNs) are hybridized with LSTM to extract the fundamental features from the input sequence automatically and construct more accurate feature representations of the investigated time series tasks. Moreover, an evolving process is introduced for the generation of the optimal configurations of the hybrid deep network by exploiting the strength of an advanced swarm intelligence (SI) algorithm, i.e. Grey Wolf Optimizer (GWO) [16].

The GWO algorithm is chosen as the candidate in this research, instead of other classical search algorithms, such as Particle Swarm Optimization (PSO) and Genetic Algorithm (GA), owing to its advantages embedded in the search mechanism. The key advantages are three-fold, i.e. the employment of multiple elite leaders, the adaptive transition from exploration to exploitation, and the stochastic nature in determining the trajectory to approach or diverge from the elicit signals [16], [17]. These advantageous characteristics endow GWO with enhanced exploration capability and search diversity, while maintaining its efficient computational cost. Comparatively, PSO is more likely to be trapped in local optima, owing to the dictation of the global best solution and the lack of diversification in its guiding signals over the entire iterative process [18], [19]. While the GA is capable of attaining the global optimality, a larger number of function evaluations are normally required. This is owing to the possibility of undermining the elicit chromosomes/solutions in the GA, as a result of the crossover operation during the evolution process [20]–[22]. Moreover, the effectiveness of GWO in terms of search efficiency over those of PSO, GA, and other similar methods [23]–[28] has been validated in many existing studies across a wide variety of problems, e.g. feature selection [29], image segmentation [30], parameter identification [31], path planning [32], and scheduling [33]. Therefore, the GWO algorithm is selected as the driving engine to search for the optimal deep neural network configurations in this research.

To be specific, an evolving GWO-based CNN-LSTM network is proposed in this research to enhance the feature representation of the time series problem as well as optimize the network topology, in order to overcome the intricate challenges of multi-variate time series prediction. We first propose a base architecture of the hybrid CNN-LSTM network, which serves as the backbone of the multi-variate time series analysis. It incorporates two convolutional layers for feature extraction, a LSTM recurrent layer for learning temporal dependencies, as well as a fully connected layer for nonlinear feature transformation. The employed convolutional layers extract the underlying granular characteristics and eliminate irrelevant factors of the input sequence automatically through various convolutional operations and nonlinear activations. As such, the temporal variations can be processed effectively and the long-term dependencies can be captured precisely by the subsequent LSTM layer owing to the more authentic feature representation.

In addition, we propose a GWO variant dedicated to the automatic search for the optimal configurations of the CNN-LSTM network. This new GWO variant aims to overcome the limitations of the original GWO model, i.e. stagnation at local optima and slow convergence rate [34]–[36], so that it is able to devise the optimal configurations of the CNN-LSTM networks efficiently. Specifically, the proposed GWO variant incorporates four distinctive strategies: 1) a nonlinear adjustment of search coefficient capable of extending the search territory during exploration and confining the search range during exploitation; 2) a chaotic weight allocation mechanism for three dominant wolf leaders using the sinusoidal chaotic map; 3) a local exploitation scheme based on an enhanced spiral search with symmetrical oscillations; 4) probability distribution-based leader enhancement. The proposed strategies intensify the search diversity by expanding the exploration space as well as diversifying the guiding signals in a periodical manner. In addition, the search efficiency and convergence rate are improved owing to the assurance of the dominance of the best leader as well as the intensified local exploitation around the optimal signals at the final stage of the search process. As such, the proposed GWO variant is capable of achieving better trade-offs between search diversification and intensification, therefore increasing the likelihood of attaining global optimality. The proposed GWO variant is subsequently employed to devise the network representation of the aforementioned CNN-LSTM model for tackling time series prediction and classification tasks. FIGURE 1 depicts the structure of the proposed GWO-based evolving CNN-LSTM time series forecasting model.

The research contributions of this study are highlighted as follows.

1. A hybrid CNN-LSTM network is proposed for time series analysis. A CNN is integrated with a vanilla LSTM to construct accurate and robust feature representations automatically, by preserving effective deterministic and stochastic trends embedded in sequence data while removing redundant and irrelevant factors.

2. Both the configuration of neural network and learning properties play an important role in model performance. The search methods employed in existing studies, such as PSO and GA, for identifying the optimal configuration of deep learning models are generally classical techniques invented decades ago. In this research, we exploit the strength of a recent SI model, i.e. GWO, for the evolution of the CNN-LSTM network. A GWO-based evolving process is devised for automatic identification of the optimal CNN-LSTM configurations in relation to the network and learning hyperparameters, i.e. the learning rate, the dropout rate, the numbers and sizes of filters in two convolutional layers, the size of pooling layer, the numbers of hidden nodes

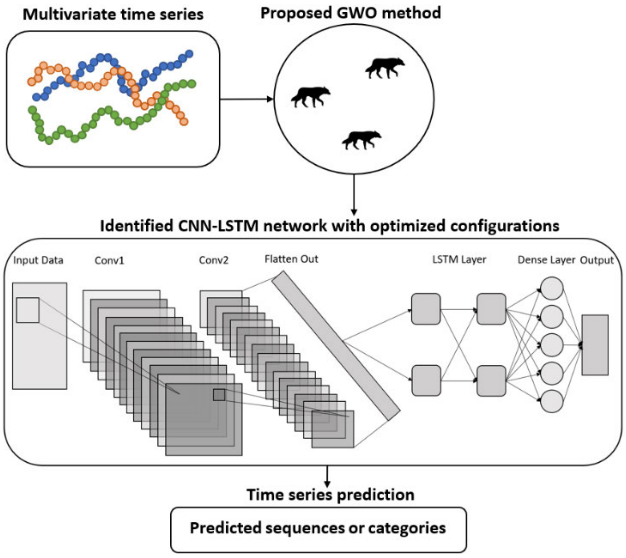


FIGURE 1. The structure of the proposed GWO-based evolving CNN-LSTM time series forecasting model where each wolf represents a set of network topology and learning hyperparameters for evolution.

in both the LSTM recurrent layer and the final dense layer, respectively.

3. Since the original GWO model is subject to stagnation at local optima and a slow convergence rate, to overcome these limitations, four distinctive strategies are proposed to enhance the search exploitation and exploration capabilities of the original model, i.e. 1) a nonlinear dynamic adjustment of search coefficient; 2) a chaotic weight allocation scheme for dominant wolves; 3) an enhanced spiral local exploitation scheme; 4) Lévy flight based leader enhancement. The proposed GWO variant is employed for automatic generation of the optimal CNN-LSTM network configuration.

4. The optimized evolving CNN-LSTM architecture is evaluated using three time series problems, i.e. energy consumption forecast, PM2.5 pollution prediction, and human activity recognition (HAR). The proposed evolving time series forecasting model outperforms those yielded by four classical SI algorithms and three advanced GWO and PSO variants on all employed time series tasks, as evidenced by statistical test results.

The rest of the paper is organized as follows. In Section II, the state-of-the-art studies on GWO and its variant models are introduced. Several recently proposed metaheuristic algorithms and the advances in related evolving deep learning models are discussed. Section III presents the details of the proposed GWO variant and the evolving CNN-LSTM network. A comprehensive evaluation of the proposed evolving CNN-LSTM time series forecasting model is provided in Section IV. Conclusions are drawn and future research directions are presented in Section V.

II. RELATED WORK

In this section, we introduce the original and modified GWO models. Several up-to-date metaheuristic algorithms are discussed. In addition, the related studies on deep learning models with hyperparameter fine-tuning and architecture generation are analysed.

A. GWO

GWO is a SI algorithm proposed recently according to the social dominant hierarchy and group hunting operations observed among grey wolves [16]. In a wolf pack, there are four different levels in terms of the positions in the social hierarchy, i.e. wolf alpha (α), wolf beta (β), wolf delta (δ), and wolf omega (ω). Those wolves from the top three hierarchies, i.e. α, β, and δ, are responsible for decision making during hunting, whereas wolves at the bottom of the hierarchical ladder, i.e. ω, are subordinates of those from the higher levels unconditionally.

In GWO, each wolf represents a randomly initialized solution. The wolves with the highest three fitness scores are labeled as α, β, and δ, respectively, and assume the leadership to guide the movement of the whole wolf pack. The GWO search scheme is based on the encircling hunting mechanism observed within the grey wolf pack in nature as well as the supposition that three dominant wolves retain better knowledge on the location of the prey (i.e., optimality) than their comrades. Henceforth, each wolf updates its position in reference to the three top leaders in the wolf pack, i.e. α, β, and δ, respectively, in a manner according to (1)-(6). The arithmetic average of the three position adjustments is then adopted as the target position for each wolf to be dispatched to, as indicated in (7).

$$D^{t+1}_{\alpha,j} = \left|C_1 \times X^t_{\alpha,j} - X^t_{i,j}\right| \tag{1}$$

$$D^{t+1}_{\beta,j} = \left|C_2 \times X^t_{\beta,j} - X^t_{i,j}\right| \tag{2}$$

$$D^{t+1}_{\delta,j} = \left|C_3 \times X^t_{\delta,j} - X^t_{i,j}\right| \tag{3}$$

$$X^{t+1}_{ad1,j} = X^t_{\alpha,j} - A_1 \times D^{t+1}_{\alpha,j} \tag{4}$$

$$X^{t+1}_{ad2,j} = X^t_{\beta,j} - A_2 \times D^{t+1}_{\beta,j} \tag{5}$$

$$X^{t+1}_{ad3,j} = X^t_{\delta,j} - A_3 \times D^{t+1}_{\delta,j} \tag{6}$$

$$X^{t+1}_{i,j} = (X^{t+1}_{ad1,j} + X^{t+1}_{ad2,j} + X^{t+1}_{ad3,j}) \Big/ 3 \tag{7}$$

$$C = 2 \times rand \tag{8}$$

$$A = (2 \times rand - 1) \times a \tag{9}$$

$$a = 2 \times (1 - \frac{t}{Max\_iter}) \tag{10}$$

where $X^t_{i,j}$ denote the element of the i-th wolf on the j-th dimension under the t-th iteration. X_α, X_β, and X_δ represent the positions of three leading wolves α, β, and δ respectively, whereas D_α, D_β, and D_δ represent the distance measures, and X_{ad1}, X_{ad2}, and X_{ad3} represent the position adjustments, in reference to the above three dominant wolves i.e. α, β, and δ, respectively. Besides that, A and C are two search coefficients related to position updating where A_1, A_2, and A_3 are the three instantiations of parameter A, and C_1, C_2, and C_3 are the three instantiations of parameter C. Max_iter denotes the maximum iteration number, whereas $rand$ is a random number in the range of [0, 1]. In addition, a denotes the exploration rate linearly decreasing from 2 to 0 as the iteration increases.

In the original GWO model, a is an essential search parameter, capable of regulating the transition from exploration to exploitation during the search iterations. The parameter a dictates the search boundary and radius of the wolf population through regulating the magnitude of the step size A, as shown in (9). Specifically, as illustrated in FIGURE 2, the wolves conduct exploration and jump out of the search range between itself and the prey when $|A| > 1$. This can only happen when the exploration rate $a > 1$, according to (9). In contrast, the exploitation between the wolf and the prey can be deployed when $|A| < 1$. As a result, the trajectory of a during the iterative process plays a significant role in affecting the exploration and exploitation capabilities of GWO. In principle, GWO possesses many merits in comparison with previous classical search methods (e.g. PSO and GA), owing to the employment of multiple-leader guided search as well as dynamic fine-tuning of the search scopes.

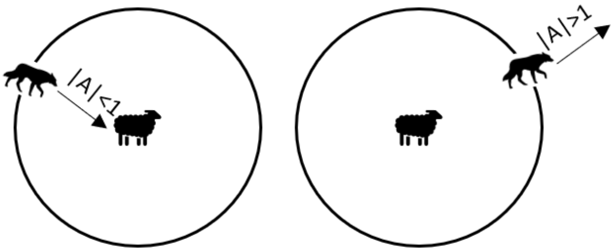


FIGURE 2. Exploitation (|A| <1) vs. Exploration (|A| >1) in GWO [16].

B. GWO VARIANTS

Despite the merits, GWO still suffers from disadvantages such as local stagnation, a slow convergence rate, as well as deficiency in fine-tuning around the best swarm leader [34]–[36]. Many efforts have been made to mitigate the identified drawbacks and enhance the search efficiency of GWO. Ozsoydan [37] proposed three GWO variants, i.e. prioGWO, learnGWO, and prLeGWO, to investigate the effects of dominant wolves in GWO. In prioGWO, three dominant wolves re-arrange their positions within themselves by following the position updating formula in the original GWO, prior to guiding the movement of the rest of the wolf pack. In learnGWO, dedicated learning curves are developed to gradually increase the dominance of wolf α, while decreasing that of wolves β and δ over the iterative process. Besides that, prLeGWO incorporates both strategies employed in prioGWO and learnGWO. These GWO variants are evaluated on multiple optimization tasks, i.e. unconstraint test functions, the uncapacitated facility location problem (UFLP), as well as the 0-1 knapsack problem. The results indicate the effectiveness of their GWO variants in comparison with those from five baseline models, i.e. PSO, GWO, a continuous PSO with a local search (CPSO), an adapted Artificial Bee Colony for binary optimization (ABCbin), and Weighted Superposition Attraction (WSA). Luo [38] proposed an enhanced GWO (EGWO) model which dynamically estimates the location of the prey using a weight-based aggregation of the three dominant wolf leaders. The weights are generated using normalized random numbers within [0, 1]. A strict hierarchical order is established by assigning the weights based on the ranking of fitness scores of the three dominant wolves, i.e. larger weights for wolves with higher rankings. Subsequently, the wolves update their positions under the guidance of this estimated location of the prey. The EGWO model is evaluated on 30-dimensional and 100-dimensional CEC2017 test functions, as well as two engineering applications. It significantly outperforms the original GWO, a fuzzy hierarchical GWO, and a random walk GWO. Gupta and Deep [39] proposed a modified GWO method based on random walks (RW-GWO). Specifically, the three dominant wolves are further improved by conducting random jumps, with the steps generated using the Cauchy distribution. RW-GWO is evaluated on 10-dimensional and 30-dimensional CEC2014 test functions. It demonstrates significant superiorities in comparison with those of the baseline models, e.g. Gravitational Search Algorithm (GSA), Cuckoo Search (CS), and Laplacian Biogeography-Based Optimization (LX-BBO).

Wang and Li [34] proposed an improved GWO (IGWO) by incorporating biological evolution and survival of the fittest principle into the evolving process of GWO. Specifically, a Differential Evolution (DE)-based breeding operation is applied to the three dominant wolf leaders. A crossover operation is then used with the yielded offspring and each individual wolf solution as the parent chromosomes. Besides that, a dynamic number of weak individuals are eliminated from the population and replaced by randomly generated new solutions, according to the principle of the survival of the fittest. The IGWO model is evaluated on the twelve benchmark functions, and it outperforms GWO, DE, PSO, ABC, and CS, statistically. Emary *et al.* [40] proposed a GWO variant, i.e. experienced GWO (EGWO). Reinforcement learning is employed in EGWO to yield the exploration rate, i.e. parameter a in GWO, for each individual wolf based on its past experience in each iteration. Specifically, a state-action model is mapped using a neural network with a single hidden layer to maximize the reward function. The input is the change state of the fitness score in every two successive iterations, and the output is the action set for the adjustment of the exploration rate, i.e. increasing, decreasing, or maintaining the current value of a. As such, the parameter a can be specifically tailored for each individual wolf by the mapped network, according to its own previous experience and performance, to bestow the freedom of choosing between exploration and exploitation on each individual wolf per se, instead of following the same regulation of parameter a collectively. The effectiveness of EGWO is evaluated on 21 feature selection tasks and 10 ANN weight adaptation tasks. The results indicate significant advantages of EGWO over the those of the original GWO, PSO and GA models. Moreover, Tu *et al.* [41] proposed a hierarchy strengthened GWO (HSGWO) model. It incorporates an elite learning operator, an opposition-based learning strategy, a DE operator, a hybrid total-dimensional and one-dimensional update strategy, as well as a perturbed operator. The enhanced elite learning strategy ensures the dominant wolves only learn from those with higher rankings, hence mitigating distractions from less advanced solutions, whereas opposition-based learning enables the dominant wolves to conduct extensive explorations. The remaining wolf solutions are able to choose between the original GWO and DE models to update their positions, in either all dimensions or only one sub-dimension. Moreover, a fraction of the wolf candidates is replaced with the solutions yielded from perturbations of randomly selected individuals from the wolf pack. HSGWO is evaluated on the CEC2014 test functions as well as 13 feature selection tasks. It outperforms the baseline models, e.g. Salp Swarm Algorithm (SSA), and differential mutation and novel social learning PSO (DSPSO).

Moreover, Gupta and Deep [42] proposed a memory-based GWO (mGWO) model. It incorporates the personal best experiences, randomly selected wolf solutions, a crossover operation, and a greedy selection strategy for position updating. The personal historical best experience is employed in two distinctive manners to yield two respective candidate solutions for the current individual under each iteration. Specifically, the first candidate is generated by replacing the position of the wolf in the current iteration with its historical best experience in the position updating equations of the original GWO algorithm. The second candidate is yielded by a local search mechanism involving the historical best experience, as well as two randomly selected wolf solutions in the neighbourhood. Subsequently, a crossover operation is performed on both candidates, and the offspring solutions are adopted as the new individuals for the next generation. Besides that, a greedy selection strategy is enforced between the wolf solutions of two consecutive iterations, and the best one is retained. The mGWO model is evaluated with the CEC2014 and CEC2017 benchmark test functions, as well as six practical engineering design problems. It outperforms numerous classical search methods, e.g. PSO, Firefly Algorithm (FA), and advanced GWO variants including Oppositional GWO (OGWO) and Improved GWO (IGWO) on unimodal, multimodal, and composite benchmark functions. Ibrahim *et al.* [43] proposed an improved GWO variant (COGWO2D) that incorporates four strategies. They are a logistic chaotic map, an Opposition-Based Learning (OBL) mechanism, a DE position updating scheme, and a disruption operator. The logistic map is used for chaotic population initialization. The OBL mechanism is applied to generate the opposite counterparts. The final collection of the initialized solutions is selected from the above combined sets according to the fitness eminence. Then, the original GWO and DE updating mechanisms are combined in parallel for position updating. In addition, the disruption operator is employed to increase the search diversity for those wolf solutions distant from the current swarm leader, while intensifying local exploitation for the remaining wolf individuals located in the vicinity of the current global best solution. Evaluated with the CEC2005 and CEC2014 benchmark functions and a feature selection task, the COGWO2D model significantly outperforms other nine competitors, including Whale Optimization Algorithm (WOA), Salp Swarm Algorithm (SSA), Ant Lion Optimizer (ALO), DE, and CS. Al-Betar *et al.* [44] investigated the impacts of different natural selection methods on the performance of GWO. In addition to the greedy selection of the top three wolf leaders employed in the original GWO model, five additional selection paradigms are explored, i.e. the tournament selection, proportional selection, stochastic universal sampling selection, linear rank selection, and random selection. Evaluated with 23 benchmark functions, GWO with the tournament selection achieves the best performances, outperforming several classical search methods, e.g. GA and PSO. GWO with the random selection obtains the worst optimization results. The research provides good insight on the common dilemma of employing elicit signals and introducing random perturbations in developing metaheuristic algorithms. Wen *et al.* [45] proposed an inspired GWO (IGWO) model. It employs a logarithmic decay function to adjust search parameter a and a modified position updating mechanism incorporating the mean position of three wolf leaders, the personal historical best experience, and the global best solution, for imitation of the position updating technique in PSO. Evaluated with four high-dimensional benchmark test functions and three practical engineering design problems, IGWO outperforms the original GWO model, four advanced GWO variants, and four other search methods. Saxena *et al.* [46] proposed a β-Chaotic map enabled GWO (β-GWO) model. It modifies the linearly decreasing search parameter a by adding a β function-based chaotic sequence. This design enables the preservation of the exploration virtue throughout the iterative process. Evaluated with the CEC2017 benchmark test functions and two practical engineering design problems, β-GWO outperforms four classical search methods, including GSA and Flower Pollination Algorithm (FPA), and five advanced GWO variants, including OGWO and Grouped GWO (GGWO), with statistical significance.

Based on the in-depth analysis of the original GWO and its variant models, we identify two major limitations of the original GWO algorithm, i.e. (i) insufficiency of exploration owing to the sharp contraction of search territory and (ii) inefficiency in the fine-tuning exploitation procedure around the global best solution, particularly in the final stage of the evolution where convergence of the population is required, owing to the distraction of the other two wolf leaders. Our literature review indicates that, instead of tackling both abovementioned problems collectively, most of the existing studies focus only on one problem, i.e. either enhancing search diversity and exploration capability via the employment of multiple position updating strategies [42], [43], [45], or improving local exploitation by reinforcing the domination of the leader (wolf α) in the leadership hierarchy [37], [38].

Comparatively, we overcome both limitations simultaneously in this research. Specifically, four strategies to enhance the original GWO algorithm are proposed, i.e.,

- a nonlinear adjustment of search coefficient capable of extending the search territory during exploration and confining the search range during exploitation;
- a chaotic weight allocation mechanism to reinforce the leadership (wolf α) while maintaining a periodic diversification of other guiding signals;
- a dedicated spiral local exploitation scheme to enhance the exploitation capability around the global best solution, in order to accelerate convergence;
- a Lévy flight-based leader enhancement scheme.

The original GWO model is enhanced from the following perspectives, i.e. a nonlinear exploration rate, chaotic diversification of guiding signals, enhanced global position updating rules, and a new spiral local exploitation mechanism. These proposed strategies work cooperatively to achieve an efficient trade-off between exploration and exploitation.

C. METAHEURISTIC ALGORITHMS

Metaheuristics are high-level algorithmic frameworks that employ generic strategies to efficiently find approximate solutions for addressing optimization problems [47]. Metaheuristic algorithms principally involve two search paradigms, i.e. exploration and exploitation. Both search paradigms are responsible for discovering a diverse assortment of solutions scattering widely across the search space, and conducting concentrated fine-tuning adjustments around promising solutions. In the research community, there are two widely accepted concepts: (1) the trade-off between exploration and exploitation is critical to the search performance of metaheuristic algorithms, and (2) a universal optimization method suitable for all problems does not exist according to ‘‘no free lunch’’ (NFL) theorem [48]. In addition to classical search methods, e.g. GA [49], DE [50], PSO [51], FA [52], MFO [53], GWO [16], GSA [54], and FPA [55], many innovative search mechanisms have been developed for further improving the robustness and applicability of metaheuristics algorithms. A review on several latest metaheuristic algorithms is presented, as follows.

Inspired by the oscillation mode and food search patterns of slime mould in nature, Li *et al.* [56] proposed a Slime Mould Algorithm (SMA). It incorporates three types of movements in cascade as well as in conjugation with oscillated search parameters for position updating. Specifically, for producing high-quality slime mould solutions, a local exploitation operation is conducted in all directions to further refine such individuals. The current low-quality positions are replaced with the new ones yielded by the global best and two other randomly selected individuals. To further increase search diversity, the slime mould population is replenished with new individuals randomly generated according to a predefined probability-based condition. Evaluated with 33 benchmark test functions and four practical engineering problems, the SMA model significantly outperforms a number of classical and advanced search methods, e.g. MFO and Comprehensive Learning PSO (CLPSO). Askari *et al.* [57] proposed a Heap-based Optimizer (HBO) by simulating various interactions in a corporate rank hierarchy. A 3-ary heap structure according to the fitness values is established on the population. A cascade search mechanism incorporating three search scenarios is developed, i.e. moving towards the immediate superior solution in the higher hierarchy (boss), moving towards a fitter solution within the same hierarchy (colleague), and retaining the current position (self-contribution). Evaluated with 97 benchmark test functions and three practical engineering problems, the HBO model outperforms seven well-known search algorithms, including Multi-Verse Optimizer (MVO), GSA, PSO, and CS. Inspired by the gradient-based Newton's method, Ahmadianfar *et al.* [58] proposed a Gradient-based Optimizer (GBO). It incorporates a gradient search rule and a local escaping operator. The gradient search rule applies a gradient-based mechanism to drive the individuals to approach the global best solution, while retaining search diversity through the employment of randomly selected individuals in the neighbourhood during the position updating process. Besides that, a local escaping operator for overcoming the local optima traps is developed by further introducing newly generated individuals into the population to participate in the competition. Evaluated with 28 benchmark test functions and six engineering problems, it significantly outperforms five classical search methods, i.e., GWO, CS, Artificial Bee Colony (ABC), WOA and Interactive Search Algorithm (ISA). Heidari *et al.* [59] proposed a Harris Hawk Optimization (HHO) algorithm. It mimics the hunting mechanism of Harries hawks. During exploration, two position updating options are developed, i.e. adjusting position in reference to the global best solution, or randomly selected solutions in the neighbourhood corresponding to two perching choices of hawks, i.e. the family member and the rabbit, during hunting, respectively. To facilitate exploitation, four local search mechanisms are designed to approach the global best solution by adopting different search coefficient vectors, in simulation of besiege processes of hawks. Evaluated with 29 benchmark test functions and six engineering optimization problems, HHO outperforms a number of classical search models, including FPA and MFO, significantly.

D. EVOLVING DEEP NEURAL NETWORKS

The performance of deep neural networks is largely dependent on the configurations of their respective architectures and hyperparameter settings. However, the search for the optimal network configuration is extremely challenging owing to the network complexity and heavy computational cost of the learning processes. Characterised with superb global search capabilities, evolutionary computation (EC) techniques have been leveraged to evolve deep learning neural networks for the identification of the optimal learning configurations as well as the discovery of innovative network structures.

Sun *et al.* [60] proposed an automatic CNN architecture design based on the GA. In this method, a generic CNN structure consisting of some predefined building blocks is employed as the foundation for the automatic architecture generation. Specifically, a building block with two convolutional layers and one skip connection is employed for the benefits of increasing the network depth without risking gradient vanishing, whereas the fully connected layers are discarded for the consideration of reducing the likelihood of overfitting resulted from the dense connection. As a result, the parameters encoded in the GA chromosomes include filter numbers of convolutional layers in each building block and the pooling layer type, with the length of chromosomes representing the network depth. The population undergoes an evolving process of the crossover operation and a mutation process. The latter incorporates four options, i.e. adding a skip layer, adding a pooling layer, removing a layer at the selected position, and changing the parameters of the building block randomly. Their proposed method is evaluated on CIFAR10 and CIFAR100 data sets. The results indicate its great superiorities in improving classification performance while significantly reducing the number of parameters, in comparison with those from the manually designed CNN models, e.g. ResNet (depth = 110), as well as the models derived from the combined schemes of automatic and manual tuning, e.g. Efficient Architecture Search (EAS) and Differential Architecture Search (DARTS). Sun *et al.* [61] proposed an evolving deep CNN (EvoCNN) model based on the GA for image classification. A variable-length gene encoding strategy is formulated to represent each potential network configuration. Two statistical measures, i.e. the mean and standard deviation values, are used to represent the weight parameters in the encoding strategy. During fitness evaluation, Gaussian distribution is employed to decode the weights based on the two statistical measures. The network architecture recommended by each chromosome as well as its corresponding decoded weights is adopted in fitness evaluation. Besides the classification performance (i.e. the mean and standard deviation of the classification error rates), the network parameter size is also considered in chromosome evaluation. A slack binary tournament selection strategy is devised for the parent chromosome selection where the mean classification performance and the parameter size are used as the threshold criteria. A unit alignment crossover operator is proposed to exchange gene information of the two parent solutions with different lengths. Evaluated with nine popular image classification data sets (e.g. Fashion, Rectangle, MNIST and its variant data sets), the EvoCNN model outperforms a number of competitive benchmark deep architectures.

Deep network generation with ResNet and DenseNet blocks based on the GA is examined by Sun *et al.* [62]. Specifically, an automatically evolving CNN (AE-CNN) model is designed to yield the CNN architectures with residual and dense connectivity. A one-point crossover operator is used for offspring solution generation, while three types of mutation operations (i.e. adding, removing, and modifying) are employed to further configure the networks. Evaluated with the CIFAR10 and CIFAR100 data sets, the AE-CNN model performs favourably as compared with a number of hand-crafted architectures and automatically devised networks from some existing methods. Despite the promising results and the great potential of the evolutionary deep learning models with respect to knowledge discovery, they are inadvertently subject to a considerably high computational cost. To overcome this drawback, Sun *et al.* [63] proposed an end-to-end performance predictor (E2EPP). A random forest is used to predict the network performance. The AE-CNN model [62] is initially employed to produce a set of CNN architectures. These network configurations are subsequently encoded into numerical decision variables, which are used in conjunction with the corresponding network accuracy rates for training the random forest-based performance predictor. Specifically, a predictor pool is generated, where each base tree model is trained using data samples containing randomly selected subsets of features. To increase ensemble robustness, a subset of base evaluators is selected to evaluate any newly created architectures based on their prediction performances with respect to the current best CNN architecture. The E2EPP model outperforms two existing performance predictors and advanced deep networks in terms of classification performance and computational efficiency.

Moreover, Martín *et al.* [64] employed a Hybrid Statistically-driven Coral Reef Optimization (HSCRO) algorithm to reconstruct the fully connected layers in VGG-16 for two purposes, i.e. reducing the amount of parameters and improving model performance. Each coral individual represents a set of fully connected layers in VGG-16. Four types of parameters are encoded in each layer, i.e. activation function, number of neurons, matrix of connection weights, and bias. The HSCRO model incorporates four evolutionary operators, i.e. asexual reproduction, sexual reproduction, settlement, and depredation, to emulate the reproduction process of coral reefs. In addition, a stratified mutation scheme is designed in which 20% of best individuals undergo parametric mutations on weights and biases, whereas the remaining 80% of individuals experience structural mutations, i.e. mutations on activation functions, the number of nodes, and node connections, during the evolving process. The identified best solution is further fine-tuned using a stochastic gradient descent (SGD) optimizer. The proposed evolving CNN model is tested on two image classification data sets, i.e. CIFAR10 and CINIC10. It is capable of reducing 90% of the connection weights while improving the classification accuracy as compared with those of the VGG-16 model.

In addition to evolving CNN models, there are studies on evolving RNN and LSTM models. Rawal and Miikkulainen [65] proposed a Genetic Programming (GP) based evolving LSTM architecture generation model. It is capable of constructing the layered network structures from a single recurrent node design. The recurrent node is encoded as a tree structure with two types activation operations, i.e. linear activations with two elements (add and multiply), and nonlinear activations with one element (tanh, sigmoid, or relu). A homologous crossover operator is designed to yield offspring solutions by crossing over the same regions of the two parent chromosomes represented in the tree structures during reproduction. Besides that, three types of mutation operations are designed for the evolution of the tree solutions, i.e. (1) replacing one activation operation with another within the same category, (2) inserting a new branch at a random position in a tree, and (3) shrinking a branch by replacing it with a randomly selected operation employed in this branch. In addition, the individual solutions with previously explored branch structures undergo repeated mutation procedures until the new tree structures are generated, in order to maintain population diversity. Two architecture generation schemes are experimented, i.e. a homogenous evolving process using a single recurrent node within a LSTM layer vs. a heterogenous evolving process using the combination of nodes with different structures. Their evolving LSTM model is evaluated in two tests, i.e. a language modelling test and an automatic music transcription test. It outperforms several existing advanced models, which include the neural architecture search method (NAS) and Recurrent Highway Network (RHN). Kim and Cho [66] developed a PSO-based evolving CNN-LSTM network for the prediction of energy consumption. The original PSO algorithm is applied to search for the optimal hyperparameters of CNN-LSTM, e.g. the filter numbers and sizes in the convolutional layers, and the number of hidden nodes in the recurrent layers, for retrieving energy consumption patterns. The results indicate that their evolving CNN-LSTM model significantly outperforms classical models, e.g. Linear Regression, Decision Tree, and Random Forest, for energy consumption prediction. Xue *et al.* [67] proposed an evolving CNN-LSTM method to tackle the inventory forecast problem. PSO and two DE variants, i.e. DE with binominal and exponential crossover operators respectively, are employed for the identification of the optimal CNN-LSTM hyperparameters, including the filter number and size in the convolutional layer, pooling type, pooling size, and stride size with respect to the pooling layer, as well as the dropout rate and the numbers of nodes in the LSTM layer and dense layer, respectively. The results indicate that the DE model with exponential crossover operator achieves the best performance in forecasting inventory. It is more advantageous for identifying proper CNN-LSTM hyperparameters in comparison with PSO as well as DE with binominal crossover operator. A systematic review on designing deep neural networks using neuro-evolution is provided in [68].

III. THE PROPOSED EVOLVING TIME SERIES PREDICTION MODEL

The proposed evolving time series prediction model consists of two major components, i.e. the CNN-LSTM network and the enhanced GWO variant. The CNN-LSTM network is the core component to make prediction based on data sequences whereas the proposed GWO variant is employed to search for the optimal hyperparameters for devising the CNN-LSTM model. In CNN-LSTM, the time series data are the inputs to the convolutional layers for it to extract the main features surrounded by the temporal context and reduce irrelevant variations. The obtained feature maps are then fed into the LSTM layers to analyze the temporal variations and learn long-term dependencies. The fully connected layer is applied subsequently to conduct nonlinear transformations on the extracted features and produce the prediction results. As discussed earlier, the performance of a deep CNN-LSTM model is significantly influenced by the quality of hyperparameter settings, such as the number of filters in the convolutional layers, the number of hidden nodes in the LSTM layer, as well as the learning configurations, e.g. learning rate, which determine the representational capacity and the training properties of the employed model. Therefore, an enhanced GWO model is proposed to automatically identify the optimal configuration of such hyperparameters for devising the CNN-LSTM network. The identified optimized CNN-LSTM model is subsequently used to undertake time series prediction and classification tasks. We explain the proposed GWO and CNN-LSTM models in detail in the following subsections.

A. THE PROPOSED GWO VARIANT

As mentioned earlier, GWO is a recently developed SI algorithm which has demonstrated robust and advanced search capabilities by the mechanism of following the guidance of top three swarm leaders, i.e. wolves α, β, and δ, as well as a dedicated design of the transition from exploration to exploitation, i.e. the exploration rate a. Despite these merits, the original GWO algorithm still suffers from severe obstacles of local optima traps, owing to its search bias, especially towards the origin of the coordinate system [38], [69], as well as the limitations of search diversity. Moreover, the static and equal division of the leadership among the three strongest wolves over the whole search course contradicts its strategy of hierarchical division within the wolf community in principle, therefore confining the capability of fine-tuning around the obtained global best solution. In this research, we propose four distinctive mechanisms to resolve the abovementioned restrictions and enhance the global exploration and local exploitation of the original GWO algorithm. Firstly, a nonlinear adjustment of the exploration rate a' is proposed to replace a and advance the search transition between exploration and exploitation. The aim is to delay the shrinkage of the search territory during exploration while concentrating the detection on the promising neighbourhood around the wolf leaders during exploitation. Secondly, a sinusoidal chaotic map is employed to generate dynamic yet clamped weights. The aim is to simulate benevolent competitions among the three dominant wolves, α, β, and δ, for leading the wolf pack. As such, a trade-off between reinforcing the leadership of the best individual and diversifying the distractions of the second and third best solutions can be achieved. Furthermore, a damped odd function with a shrinking amplitude is proposed. The aim is to deploy a fine-tuning local search process around the swarm leader in the final stage for accelerating the convergence process. Lastly, the Lévy flight is employed with the aim to further enhance the quality of three leading wolves α, β and δ, in each iteration, in order to overcome early stagnation. The pseudo-code of the proposed GWO variant is provided in **Algorithm 1**.

Algorithm 1 The Proposed GWO Model

```
Start
Initialize a grey wolf population
Evaluate each individual using the objective function $f(x)$
  and identify three dominant wolves with the best fitness
  scores, denoted as $X_\alpha$, $X_\beta$, and $X_\delta$, respectively
While ($t < Max\_iter$)
{
    Update the exploration rate $a'$ by (11)-(12)
    Generate dominance factors for three wolf leaders,
      i.e. $w$ for wolf $\alpha$ and $w'$ for wolves $\beta$ and $\delta$,
      using (17)-(18)
    For (each leader) do
    {
        Conduct leader enhancement using Lévy
        flight as defined in (24)
    } End For
    If ($t < 0.8 \times Max\_iter$)
    {
        For (each wolf $i$ in the population) do
        {
            Generate step size $A'$ using (13)
            Calculate distance measures, $D_\alpha$, $D_\beta$, and $D_\delta$, by (1)-(3)
            Update the position with respect to
              $X_\alpha$, $X_\beta$, $andX_\delta$, by (14)-(16), & (19)
        } End For
    Else $t \geq$ Ý$0.8 \times$ Á$Max\_iter$
        For(each wolf $i$ in the population) do
        {
            Conduct local exploitation around the
            best leader $X_\alpha$ with dynamic steps by
            (20)-(22)
        } End For
    } End If
    For(each wolf $i$ in the population) do
    {
        Calculate the fitness score of $i$
        Update three dominant leaders $X_\alpha$, $X_\beta$, $andX_\delta$
    } End For
} End While
Output the most optimal solution $X_\alpha$
End
```

1) A NONLINEAR EXPLORATION FACTOR FOR ADJUSTMENT OF THE SEARCH BOUNDARY

In the original GWO algorithm, the transition from exploration to exploitation is governed by the exploration rate a, as defined in (10). It decreases linearly from 2 to 0 as the iteration builds up. This linear changing pattern largely governs the search performance, owing to the lack of distinction among different search behaviours during exploration and exploitation. To be specific, the search parameter a determines how far each individual wolf can jump with reference to the leader wolves. This is achieved by manipulating the magnitude of step size A, where $A| \leqslant |a|$ is always satisfied throughout the search process, as in (9). As discussed earlier, a is the determining factor that controls the boundary of the search territory. The linearly decreasing pattern of a in the original GWO algorithm results in an acute shrinkage of the search territory during exploration as well as lack of search attention on the promising vicinity of wolf leaders during exploitation. In fact, a number of existing studies have explored adaptive control of the diverse search operations based on a variety of nonlinear functions, e.g. trigonometric [70], [71], exponential [72], [73], and logarithmic-based functions [45], [74], which have produced impressive performances. Inspired by the existing studies, we propose a nonlinear search parameter a' which integrates trigonometric functions, i.e. *cos* and *sin*, as well as the hyperbolic function, i.e. *tanh*, to overcome the limitations of a in the original GWO algorithm. The aim is to alleviate the sharp contraction of the search territory in the early search stage of the original GWO algorithm as well as to achieve a superior transition from exploration to exploitation. The formulae of the newly proposed exploration factor a' are presented in (11)-(12).

$$a' = 2 \times \left(\cos\left(\frac{(tanh\theta)^2 + (\theta \sin \pi\theta)^k}{(tanh1)^2} \times \frac{\pi}{2}\right)\right)^2 \tag{11}$$

$$\theta = \frac{t}{Max\_iter} \tag{12}$$

where t and Max_iter represent the current and the maximum numbers of iterations, respectively, while θ is the quotient of t divided by Max_iter. The coefficient k determines the descending slope of the search parameter a' over the search process, therefore capable of regulating the transition from exploration to exploitation. Based on trial-and-error, $k = 5$ is adopted in this research. FIGURE 3 presents a plot of the proposed nonlinear exploration rate a', against the linearly decreasing a adopted in the original GWO algorithm as defined in (10).

The proposed nonlinear search parameter a' replaces a in the original GWO algorithm to generate step size A' for the movement of an individual wolf with respect to each wolf leader, i.e. α, β, and δ, as shown in (13)-(16). Based on the new step size A', the movement mechanism towards each wolf leader in the original GWO model is performed, as defined

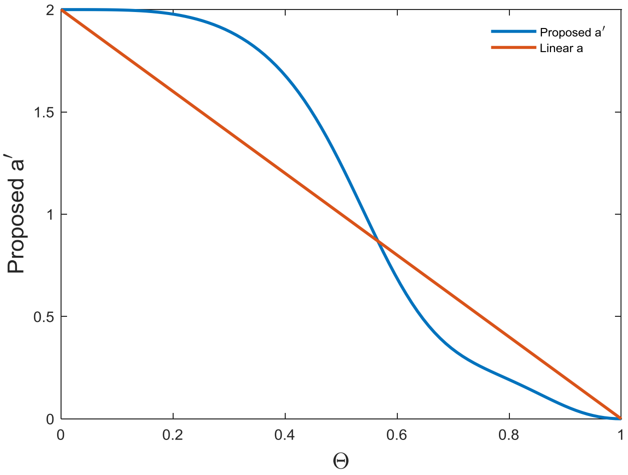


FIGURE 3. The proposed nonlinear *a*′ *vs.* linear *a* in the original GWO.

in (1)-(6).

$$A' = (2 \times rand - 1) \times a' \tag{13}$$

$$X_{ad1,j}^{t+1} = X_{\alpha,j}^{t} - A'_1 \times D_{\alpha,j}^{t+1} \tag{14}$$

$$X_{ad2,j}^{t+1} = X_{\beta,j}^{t} - A'_2 \times D_{\beta,j}^{t+1} \tag{15}$$

$$X_{ad3,j}^{t+1} = X_{\delta,j}^{t} - A'_3 \times D_{\delta,j}^{t+1} \tag{16}$$

where A' is the step size yielded by the proposed search parameter a'. Three step sizes, i.e. A'_1, A'_2, and A'_3, are generated for the movements towards three dominant wolves, i.e. α, β, and δ, respectively, for each individual wolf under position updating. In addition, X_{ad1}, X_{ad2}, and X_{ad3} denote the position adjustments with respect to α, β, and δ, respectively, while $D_{\alpha,j}^{t+1}$, $D_{\beta,j}^{t+1}$ and $D_{\delta,j}^{t+1}$ are obtained using (1)-(3).

As shown in FIGURE 3, in comparison with the linear adjustment of a in the original GWO algorithm, the proposed nonlinear exploration factor a' decreases with gentle gradients both at the beginning and the end of the search process. In other words, the nonlinear gradient variation pattern of the proposed a' is capable of yielding significantly larger exploration rates in the first half of the search course, as well as smaller exploration rates in the second half of the search course. As a result, the search boundary can be maintained at a level with only minor contraction, and the search territory can be significantly expanded during exploration. At the same time, the local detection procedure is focused on the vicinity of the promising solutions, owing to the confined search boundary during exploitation. These advantages are strengthened when deploying a' to the movement of an individual wolf towards each of the three dominant wolves (α, β, and δ), therefore enhancing both search diversification in exploration and search intensification in exploitation. As such, an enhanced transition from exploration to exploitation can be achieved by using the proposed nonlinear exploration rate a', in comparison with that of the linearly decreasing parameter a in the original GWO algorithm.

2) CHAOTIC DOMINANCE OF WOLF LEADERS

In the original GWO algorithm, although motivated by the social hierarchy observed among grey wolves, the leadership within the wolf pack is evenly divided and assumed by three dominant leaders. This arrangement remains static over the whole iteration course, regardless of the difference of the fitness scores of the wolf leaders. This lack of prioritizing operators among the dominant wolf leaders results in a slow convergence rate, therefore compromising search efficiency [34], [75]. Motivated by many diverse strategies proposed to establish dynamic and strict social leadership hierarchies in GWO, e.g. dedicated learning curves [37] and assignment of random weights according to the fitness scores [38], we employ a sinusoidal chaotic map to generate the weight factors for prioritizing the dominance of the best leader wolf α, as shown in (17). Then, the leadership factors of wolves β and δ are determined subsequently in accordance with that of wolf α, as indicated in (18). The position updating mechanism with the new dominance factors is presented in (19).

$$w_{t+1} = 2.3 \times w_t^2 \times \sin(\pi w_t) \tag{17}$$

$$w'_{t+1} = 0.5 \times (1 - w_{t+1}) \tag{18}$$

$$X_i^{t+1} = w_{t+1} \times X_{ad1}^{t+1} + w'_{t+1} \times X_{ad2}^{t+1} + w'_{t+1} \times X_{ad3}^{t+1} \tag{19}$$

where w_t and w_{t+1} represent the weight coefficients of the position adjustment X_{ad1} with respect to wolf α in the t-th and $(t+1)$-th iterations, respectively, while w'_{t+1} represents the weight coefficient for both position adjustments X_{ad2} and X_{ad3} with respect to wolves β and δ, respectively, in the $(t+1)$-th iteration.

The proposed chaotic dominance scheme is capable of achieving a better trade-off between reinforcing the leadership of the best wolf solution (single-leader guided search) and diversifying the guiding signals (multi-leader guided search). As illustrated in FIGURE 4, the employed sinusoidal chaotic map produces dynamic values roughly within the range of [0.5, 0.9], which are adopted to represent the irregular characteristic of the leadership of the most dominating wolf α. The proposed leadership assignment scheme simulates a centralized wolf regime in which wolf α is bestowed with the highest authority. The leadership assumed by wolf α is greater than the combined power of wolves β and δ. As a result, the search procedure becomes more focused on promising territories represented by wolf α, mitigating the negative impacts of malignant distractions and futile movements caused by less promising leader signals. As such, the convergence speed becomes faster and the search efficiency improves.

In addition, the chaotic map oriented dynamic dominance of wolf α increases search diversity by diversifying the guiding signals, in comparison with the static and equal leadership operation employed in the original GWO method. Specifically, as the weight coefficients fluctuate periodically between [0.5, 0.9], the dominance level of wolf α varies accordingly over the whole iterative process. The rivalry from

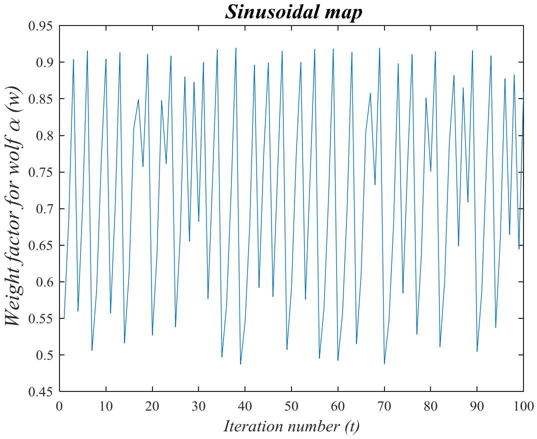


FIGURE 4. The sinusoidal chaotic map used for generating the leadership factors of the most dominating wolf α.

wolves β and δ intensifies and becomes equivalent to that of wolf α when the weight coefficient produced by the chaotic map is equivalent to 0.5. As a result, the distraction imposed by wolves β and δ can effectively dilute the dominance of wolf α and divert the undergoing search trajectory to an unexploited new region. As shown in FIGURE 4, such drastic changes in leadership assumptions occur more frequently in the middle of the search process, i.e. between 30-60 iterations. This phenomenon can effectively prevent the wolf pack from being trapped in local optima and reduce the likelihood of premature stagnation.

Moreover, the employed dynamic rivalry of the dominance among three leading wolves assimilates the merits from both multi-leader and single-leader guided search procedures. Specifically, the significant dominance of wolf α, induced by the relatively larger weight coefficient w_{t+1} as indicated in (17), enables the enhanced GWO model to emulate the efficiency of single best-leader guided search. On the other hand, the equivalent rivalry from wolves β and δ, induced by a comparatively smaller weight coefficient w'_{t+1} as defined in (18), allows the proposed GWO model to leverage the strength of global exploration from multi-leader guided search. In contrast, the existing studies [37], [38] in reinforcing the leadership of wolf α generally fail to consider the influence of the confrontation from the perspective of the combined power of wolves β and δ. Indeed, the lack of variance in leadership contention in the existing studies also increases the risk of local stagnation.

Overall, the proposed chaotic leadership assignment among the elite wolf circle in conjunction with the nonlinear adjustment of the search boundary enables the enhanced GWO model to achieve more efficient trade-offs between exploration and exploitation from two levels, i.e. the independent movement with respect to each wolf leader, and the aggregation pertaining to the three wolf leaders.

3) A DEDICATED LEADER EXPLOITATION SCHEME

The constant adherence to the guidance of three best wolves through the whole iterative process propels the search diversity of GWO. On the other hand, it also constrains the capability of concentrating on local detection around the identified best solution. We subsequently propose a damped function with decremental amplitudes to produce a variety of step sizes for local exploitation and fine-tuning around wolf α at the final search stage (e.g. $t \geq 80$), as well as to guarantee convergence of the wolf population. The damped function is illustrated in (20), whereas the position updating equation based on the generated step size is presented in (22).

$$\lambda = f \times e^{3r^2/2} \times \cos(\pi r) \times \sin(\pi r) \tag{20}$$

$$f = 1 - 0.05 \times (t - 80) \tag{21}$$

$$X_{i,j}^{t+1} = X_{\alpha,j}^{t} - \lambda \times \left| X_{\alpha,j}^{t} - X_{i,j}^{t} \right| \tag{22}$$

where λ and f denote the yielded step size and amplitude of the damped function, respectively, while $X_{i,j}^{t+1}$ represents the element of wolf i at the j-th dimension in the $(t+1)$-th iteration. Besides that, r is a random value in the range of $[-1, 1]$, and X_α denotes the position of the best wolf leader α.

As shown in FIGURE 5, the proposed formula is an odd function with damped oscillations along the x axis. When x is in the clamp between $[-1, 1]$, the range of the highest crest and trough is $[-1.3, 1.3]$, whereas that of the second highest crest and trough is $[-0.6, 0.6]$. As a result, the wolf solutions are capable of conducting large jumps from all directions radiated from wolf α when $|r| > 0.5$, as well as performing granular movements when $|r| < 0.5$. Moreover, the symmetry of the function with respect to the coordinate origin induces an even distribution of the generated steps in both the positive and negative realms. This enables the simulation of individual wolves to approach wolf α as well as to distance from it with an equal probability. Furthermore, a decremental amplitude f is applied to gradually flatten the fluctuation and shrink the search radius as the iteration builds up. The intensification of the detection around the best solution is, therefore, strengthened through this dedicated local exploitation scheme.

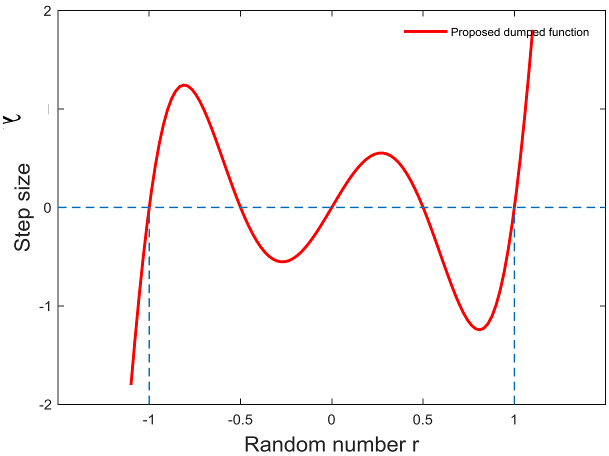


FIGURE 5. The proposed damped function in (20) when $f = 1$.

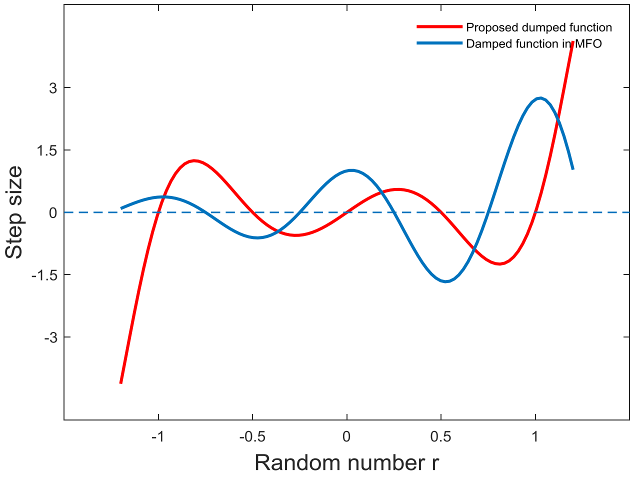


FIGURE 6. **The comparison between proposed damped function and the damped function applied in MFO.**

As depicted in FIGURE 6, we further compare the proposed formula in (20) with the damped function employed in the spiral search mechanism of MFO [53] defined in (23).

$$y = e^{br} \times \cos(2\pi r) \tag{23}$$

where b is a constant and is set as 1, while r is a random value in the range of $[-1, 1]$. Besides that, y is the yielded step size.

Firstly, the damped function in MFO does not possess any symmetrical properties. Secondly, it does not involve any dynamic granular changes in its search scale. As a result, the variance of the oscillated scales and the imbalance of the probabilities between generating identical (positive values) and reverse (negative values) search directions can lead to obstinate search bias and incomplete coverage of the search territory, which could degrade search efficiency and local intensification. In contrast, the proposed strategy is able to effectively accelerate convergence as well as intensify local exploitation around the identified best leader, owing to the increased diversity in terms of scales and symmetric directions of the search steps.

Moreover, the switch condition, i.e. $t \geq 0.8 \times Max\_iter$, in **Algorithm 1**, is adopted based on trial-and-error to fully unleash the potential of exploration in the proposed GWO model while ensuring sufficient time window for executing the proposed local exploitation scheme. Such a control mechanism also enables the proposed algorithm to take advantages of both local and global search mechanisms to address the limitations of the original GWO method, and achieve an efficient trade-off between diversification and intensification. Specifically, if this threshold is too small, the search is likely to stagnate at local optima owing to the premature transformation from the multi-leader guided global exploration into the single-leader based local detection. If the threshold is too large, the algorithm is likely to suffer from insufficient local exploitation around the global best solution, as in the case of the original GWO model, owing to the distraction of the other two wolf leaders (β and δ) as well as the narrow window left for executing the rectified spiral fine-tuning operation. Therefore, based on trial-and-error, we set 0.8 as the threshold, which offers an efficient trade-off between global exploration and local exploitation to increase the likelihood of attaining global optimality.

4) WOLF LEADERS ENHANCEMENT USING LÉVY FLIGHT

The quality of the dominant leaders is crucial to the performance of GWO, owing to the adoption of multiple leaders in the search process. We, therefore, implement a Lévy flight random walk as defined in (24) to further improve the quality of the three leading wolves successively.

$$X'_{L,j} = \begin{cases} X_{L,j} + \xi \times X_{\sigma,j} & \textit{if rand} > 0.5 \\ X_{L,j} & \textit{otherwise} \end{cases} \tag{24}$$

where X_L and X'_L represent the positions of each wolf leader before and after performing a random walk according to the Lévy distribution, respectively. X_σ represents a distinctive second wolf leader selected among α, β and δ as a distraction signal, while ξ denotes the step size generated from the Lévy distribution [76].

The Lévy jumps are only implemented on the dimensions where the determinants are higher than 0.5. Only the mutated offspring solutions with improved fitness scores are retained. For each leader undergoing mutation, a second distinctive dominant leader is randomly selected and employed to introduce the distinguishing factors. This distraction from a different leading wolf can effectively prevent the vanishing of the jump momentum resulted from stagnation at local optima located next to the coordinate origin, i.e. $X_{L,j} = 0$. In short, this leader enhancement operation based on Lévy flight enables the wolf pack to jump out of local optima traps and increases the likelihood of attaining global optimality.

Overall, the proposed GWO variant employs four strategies to enhance search diversity while accelerating convergence, i.e. a nonlinear adjustment of search boundary, a chaotic dominance rivalry among leading wolves, a dynamic leader exploitation operation using an enhanced spiral search procedure, as well as a Lévy flight mutation operation based on the dominant wolves. These proposed strategies enhance the original GWO algorithm from three perspectives, i.e. adjusting the search parameters, modifying the position updating rules and search processes, and enhancing the promising leader signals. These strategies work cooperatively to mitigate premature convergence, improve the transition from exploration to exploitation, and overcome the limitations of the original GWO model.

B. THE PROPOSED CNN-LSTM ARCHITECTURE

CNN-LSTM has attracted many research attentions owing to its great advantages in combining the strength of automatic feature extraction in CNN and the capability of capturing long-term temporal dependencies in LSTM. The convolutional layer in CNN-LSTM disentangles the cross-correlations while preserving deterministic and stochastic trends embedded among the input time series. Therefore, it produces more accurate feature representations, which enables the LSTM layers to learn temporal dependencies more precisely. The CNN-LSTM networks have been applied to tackle a variety of time series prediction and classification problems successfully, e.g. stock market forecasting [77], named entity recognition [78], textual sentiment analysis [79], [80], machine translation [81], facial expression recognition [82], and image description generation [83].

In this research, we propose a skeleton architecture of CNN-LSTM, upon which the tailored configuration of the hyperparameters is set according to the recommendation of the proposed GWO variant with respect to the investigated time series tasks. The topology of the proposed CNN-LSTM architecture is outlined in FIGURE 7. It consists of three core types of layers, i.e. the convolutional layer, the LSTM recurrent layer, and the dense layer.

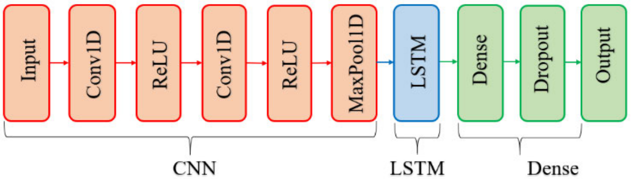


FIGURE 7. The topology of the proposed CNN-LSTM architecture.

The input data sequence is firstly used as the input to two consecutive convolutional layers for feature extraction. Through the convolutional operations of filters with different properties, nonlinear activation of neurons, and abstract representation of max pooling, the low-level features and distinctions among variables under the context of temporal effects are, therefore, acquired. The obtained feature map is then passed to the LSTM layer, where the complex dependencies are thoroughly learned by the examination of three effective gates in LSTM, i.e. the forget, input, and output gates. Specifically, the irrelevant or redundant information from previous cell states is removed by the forget gate. The effective new information from the input sequence is stored by the input gate. Moreover, the signals from the cell state are filtered and then propagated to the next state by the output gate. Furthermore, the processed temporal information is used as the input to the dense layers to undergo nonlinear transformation. Finally, the obtained information is projected to the output space, and the prediction results are produced. Overall, the proposed CNN-LSTM skeleton architecture is adopted as the foundation for evaluating the time series problems in various test scenarios.

C. THE PROPOSED GWO-BASED EVOLVING CNN-LSTM NETWORK

The identification of the optimal configurations of hyperparameters and architectures is crucial to the performance and efficiency of deep neural networks in practice. Such configuring and searching processes are particularly cumbersome for CNN-LSTM owing to the increased number of hyperparameters induced by the hybridisation of CNN and LSTM. The complexity is increased by the profound interactive effects among the hyperparameters of both CNN and LSTM, in comparison with those from monotonous deep learning models. In this study, we employ the proposed GWO model for automatic optimal configuration identification of the CNN-LSTM architecture, to undertake time series prediction tasks.

To be specific, the proposed GWO variant is employed to automatically search for the optimal hyperparameters and topologies of the CNN-LSTM model, by optimizing a series of learning and network parameters, i.e. the learning rate, the dropout rate, the numbers and sizes of filters in two convolutional layers, the size of the pooling layer, as well as the numbers of hidden nodes in the LSTM recurrent layer and the final dense layer. The search range of each optimized parameter is presented in TABLE 1. The explored hyperparameters include the key factors critical to the representational capacity of CNN-LSTM, which include the number of hidden nodes in the LSTM layer and those responsible for the learning efficiency and training property, e.g. the learning and dropout rates. As such, the confounding effects and impacts of various hyperparameters can be thoroughly explored through the evolving process of the proposed GWO variant. The CNN-LSTM model with the identified optimized configuration is then applied to tackle time series prediction and classification tasks.

TABLE 1. The search range of the hyperparameters.

Optimized component	Hyperparameter	Range
Conv	No. of filter in 1st layer filter size in 1st layer No. of filter in 2nd layer filter size in 2nd layer	$[2^0, 2^{10}]$ [1, 5] $[2^0, 2^{10}]$ [1, 5]
Pooling	pooling size	[2, 5]
LSTM	No. of hidden nodes	[10, 500]
Dense	No. of nodes	[10, 200]
Learning configuration	learning rate dropout rate	$[10^{-5}, 10^{-1}]$ [0, 0.6]

The optimal hyperparameter search of CNN-LSTM is performed as follows. Firstly, the population of the proposed GWO variant is randomly initialized. Each individual (wolf) represents a possible configuration of the optimized CNN-LSTM model. Based on the training data set, the recommended CNN-LSTM model with the specific structure and parameter settings represented by each wolf is trained. Based on the validation set, the fitness scores, i.e. the error rate for the classification problems or the root mean square error (RMSE) for the regression problems, are computed. The solutions with the top three fitness scores are identified as the dominant wolves, and are employed to guide the entire wolf population to search for the global optimality by following the proposed GWO algorithm. The optimal configuration obtained by the best wolf leader is adopted to yield the finalised CNN-LSTM model. It is re-trained using the combined training and validation sets and then evaluated using the unseen test data set. The pseudo-code of the proposed evolving CNN-LSTM model is provided in **Algorithm 2**. In our empirical studies, we employ several benchmark time series problems to examine the effectiveness of the proposed GWO-based CNN-LSTM model. The detailed results and analysis are presented in Section IV.

Algorithm 2 The Proposed Evolving CNN-LSTM Network

```
Start
Initialize a grey wolf population with each individual
  representing a specific network configuration of
  CNN-LSTM
Prepare training, validation and test sets
For (each wolf $i$ in the population) do
{
    Decode $i$ into the corresponding CNN-LSTM network
    Train the network using the training set
    Evaluate the network on the validation set and calculate
    the fitness score
} End For
Identify three dominant wolves (denoted as $X_\alpha$, $X_\beta$, and $X_\delta$)
  with the best fitness scores
While ($t < Max\_iter$)
{
    Evolve the wolf population according to the search
    mechanism of the proposed GWO method, i.e. line
    6-30 in Algorithm 1
} End While
Output the most optimal solution $X_\alpha$
Decode $X_\alpha$ into the corresponding CNN-LSTM network
Train the identified optimized network on the combined
  training and validation sets
Evaluate the optimized CNN-LSTM network on the test
  set and output the test result
End
```

TABLE 2. Parameter settings of search methods.

Methods	Parameter settings
GWO [16]	step size $A = (2 \times rand - 1) \times a$, where a linearly decreases from 2 to 0, $rand \in (0, 1)$. search parameter $C = 2 \times rand$.
PSO [51]	cognitive component $c_1 = 1.4962$, social component $c_2 = 1.4962$, inertia weight $w = 0.7298$.
GSA [54]	initial gravitational constant $G_0 = 100$, search parameter $\alpha = 20$.
FPA [55]	switch probability = 0.8, step size L for global pollination drawn from a Lévy flight distribution, step size ϵ for local pollination drawn from a uniform distribution within [0, 1].
CSO [85]	r_1, r_2, and r_3 are search parameters randomly drawn from a uniform distribution within [0, 1].
PrLe GWO [37]	initial weights of three dominant wolves $w_\alpha = 1/3$, $w_\beta = 1/3$, and $w_\delta = 1/3$, weights of three dominant wolves at the end of the iteration $w_\alpha = 0.8$, $w_\beta = 0.1$, and $w_\delta = 0.1$.
Fuzzy GWO [84]	A Mamdani fuzzy system to generate weights for three dominant wolves.
Prop. GWO	A nonlinear exploration factor: $a' = 2 \times \left(\cos\left(\frac{(tanh\theta)^2 + (\theta \sin \pi\theta)^5}{(tanh 1)^2} \times \frac{\pi}{2}\right)\right)^2$, where θ is the quotient of the current iteration number divided by the maximum iteration number. A step size for leader exploitation: $\lambda = f \times e^{3x^2/2} \times \cos(\pi r) \times \sin(\pi r)$, where f linearly decreases from 1 to 0, while r is a random number in [-1, 1]. Leader dominance coefficient generation using the sinusoidal map.

IV. EVALUATION AND DISCUSSION

In this section, the proposed evolving CNN-LSTM model is evaluated on two time series prediction problems, i.e. building energy consumption forecast and PM2.5 concentration prediction, and one time series classification problem, i.e. human activity recognition. The performance of the proposed GWO variant in identifying the optimal CNN-LSTM configurations is compared against those of four classical search methods, i.e. GWO [16], PSO [51], GSA [54], and FPA [55], as well as three advanced GWO and PSO variants, prLeGWO [37], FuzzyGWO [84], and CSO [85]. The parameter settings of the baseline models are provided in TABLE 2. The following settings are employed for each experiment to ensure a fair comparison, i.e. the maximum number of function evaluations = population size (30) × the maximum number of iterations (100). A CNN-LSTM model with the default parameter settings, i.e. filter number in the 1st Conv layer = 32, filter size in the 1st Conv layer = 2, filter number in the 2nd Conv layer = 32, filter size in the 2nd Conv layer = 2, pooling size = 2, number of node in LSTM layer = 300, number of node in dense layer = 100, learning rate = 0.001, and dropout rate = 0.2, is also employed as one of the baseline models for performance comparison. We conduct our experiments using a Tesla K80 GPU with 12 GB RAM. Moreover, we conduct ten independent runs for each experiment to mitigate the impact of random factors on the evaluation. The experimental details of the employed time series prediction problems are presented, as follows.

A. ENERGY CONSUMPTION FORECAST

1) DATA SET

First of all, the individual household electricity consumption data set[1] from the UCI machine learning repository [86] is employed to evaluate the effectiveness of the proposed evolving CNN-LSTM model. The data set contains 2,075,259 measurements with nine attributes collected in an interval of one minute, from a house located in Sceaux between December 2006 and November 2010.

[1] The URL of the household electricity consumption data set is: https://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption.

2) EXPERIMENTAL SETTINGS

According to the difference of the time interval, energy forecasting models are generally classified into three categories, i.e. short-term, medium-term, and long-term energy forecast [87]. In this research, we develop a multi-input and multi-output short-term energy forecasting model. Specifically, we predict the amount of electricity consumption for the next week using the historical data from the previous two weeks, in order to capture weekly periodicity and irregularity of the energy consumption. The proposed weekly energy forecasting model can be used to inform future energy expenditures of the household, and to facilitate the demand management. The original observations with an interval of one minute are transformed into daily energy consumption data for the weekly prediction of energy consumption. We employ the data samples from the first two years for training, while those from the subsequent one year for validation and from the final year for test.

We optimize eight of the total hyperparameters listed in Table 1 except for the pooling size, for the prediction of energy consumption. The pooling size is set to 2, owing to the comparatively small input vector of the sequential data of this energy consumption scenario, i.e. 14 × 9, where 14 and 9 represent time steps and the feature size, respectively. The optimal CNN-LSTM configuration is identified based on the training and validation sets. The batch size is set to 128, whereas a total of 20 epochs are used in the training stage to balance between performance and computational cost. In addition, the Adam optimizer is applied in the training process while the RMSE is adopted as the fitness score to evaluate the performance of CNN-LSTM. The devised CNN-LSTM model is retrained on the combined set of training and validation samples for 100 epochs. Finally, the fully trained CNN-LSTM model is employed to forecast energy consumption on the unseen test set.

3) RESULTS AND DISCUSSION

Two performance indicators are employed to evaluate the effectiveness of the proposed evolving CNN-LSTM model, i.e. RMSE and the mean absolute error (MAE). The respective results over ten independent runs are presented in TABLE 3 and TABLE 4.

TABLE 3. The RMSE results over 10 independent runs.

Run	CNN-LSTM	GWO	PSO	GSA	FPA	CSO	prLe GWO	Fuzzy GWO	Prop. GWO
1	401.7	403.3	392.4	418.0	368.0	396.9	388.2	384.4	371.2
2	409.6	413.4	395.6	425.0	403.9	376.0	385.1	433.4	383.0
3	451.1	415.3	410.6	401.7	541.6	483.2	418.2	413.1	382.5
4	421.0	441.3	387.7	383.3	400.9	406.4	381.8	369.6	365.0
5	422.1	412.0	399.4	408.2	437.4	506.7	404.1	398.9	376.9
6	439.2	423.3	381.5	424.6	421.9	381.2	393.1	375.9	386.5
7	424.1	396.7	397.0	426.2	391.4	410.6	387.2	378.1	376.7
8	419.6	383.5	384.8	390.0	382.4	394.7	383.4	400.1	380.3
9	418.8	383.6	400.2	438.8	391.9	406.1	377.4	407.8	382.6
10	415.6	483.4	395.7	429.6	381.8	387.6	379.6	401.0	367.1
Avg.	422.3	415.6	394.5	414.5	412.1	414.9	389.8	396.2	**377.2**

TABLE 4. The MAE results over 10 independent runs.

Run	CNN-LSTM	GWO	PSO	GSA	FPA	CSO	prLe GWO	Fuzzy GWO	Prop. GWO
1	310.5	303.0	299.6	314.5	280.1	313.3	294.2	291.8	287.2
2	316.6	321.4	305.9	320.9	313.0	287.3	294.7	340.7	292.5
3	345.2	318.8	322.5	307.3	419.4	367.8	328.2	320.0	292.7
4	319.2	346.5	286.6	294.3	306.3	309.2	291.3	286.5	277.7
5	326.7	321.7	305.9	301.9	335.8	365.0	307.4	306.6	291.5
6	342.9	321.0	286.5	310.7	326.2	294.8	302.4	289.9	299.4
7	324.4	307.1	300.0	324.4	297.9	314.1	286.6	291.1	296.4
8	322.4	284.0	298.4	301.1	298.1	303.5	289.1	314.0	290.1
9	324.6	297.6	313.8	328.9	302.0	310.4	294.6	316.6	294.7
10	312.8	379.4	302.6	326.6	292.4	298.9	291.8	316.5	283.1
Avg.	324.5	320.1	302.2	313.1	317.1	316.4	298.0	307.4	**290.5**

The optimized CNN-LSTM networks identified by the proposed GWO variant achieve the lowest RMSE and MAE results and demonstrate significant advantages in comparison with those yielded by four classical search methods and advanced prLeGWO, FuzzyGWO, and CSO models, as well as the CNN-LSTM network with the default setting. As shown in TABLE 3, the RMSE results produced by the proposed GWO-based evolving CNN-LSTM model are more reliable, lying within the range of [360, 390], whereas most of the RMSE results produced by the baselines methods are larger than 390, demonstrating greater variances. As shown in TABLE 4, the significant superiorities of the proposed GWO model can also be observed from the MAE results. This indicates that the optimized CNN-LSTM configurations identified by the proposed GWO variant are capable of identifying spatial variations among time series variables and extracting irregular patterns in temporal information embedded in the energy usage data, effectively.

We further analyze the advantages of the proposed GWO method by examining the distinctive characteristics of its identified CNN-LSTM configurations, as opposed to those yielded by the baseline models. The mean hyperparameters of the optimized configurations of CNN-LSTM yielded by the GWO variant over 10 runs are presented in TABLE 5. In general, the CNN-LSTM structures identified by the proposed GWO model exhibit two main distinctive characteristics, i.e. a higher number of filters in the first convolutional layer and a moderate setting of the numbers of nodes in the recurrent and dense layers, in comparison with those identified by

TABLE 5. The mean configurations of the identified cnn-lstm networks over 10 runs.

Conf.	GWO	PSO	GSA	FPA	CSO	prLe GWO	Fuzzy GWO	Prop. GWO
No. 1st	82.8	118.8	63.2	110.6	186.6	12.4	105	217.2
No. 2st	4.6	17	8.4	159	22.6	11.8	5.8	5.4
S. 1st	3.6	3.7	2.9	3.1	2.2	2.0	1.9	1.6
S. 2st	1.8	1.8	2.9	2.1	2.7	1.3	1.3	1.2
LSTM	327.3	320.6	261	246	321.8	122.9	129.5	284.1
Dense	70.5	110.3	96.6	74.9	68.7	55.5	16.8	35.4
DR	0.246	0.270	0.336	0.312	0.341	0.193	0.041	0.170
LR	0.024	0.042	0.051	0.034	0.051	0.023	0.003	0.021

the baseline models. Specifically, the optimized CNN-LSTM structures are capable of extracting energy usage features more effectively owing to the higher number of filters in the first convolutional layer, i.e. 217.2. These filters in the convolutional layer are able to reduce noise and remove irrelevant variations among time series variables while preserving the essential temporal variance. Besides that, the long-term dependencies can be acquired efficiently without overfitting, owing to the optimized and more balanced settings of the hidden nodes in the LSTM and dense layers, i.e. 284.1 and 35.4, respectively. As such, the devised CNN-LSTM networks are capable of achieving more efficient trade-offs between the model representational capacity and the avoidance of overfitting.

In contrast, the network configurations yielded by the baseline methods and default CNN-LSTM model generally achieve inferior learning capacities in incorporating temporal information with respect to the energy usage patterns, owing to the lack of the convolutional operations and sub-optimal recurrent network representations. This indicates the deficiency of the baseline search methods in exploring sophisticated interactions among hyperparameters in CNN-LSTM. In other words, the baseline models are more prone to local optima traps, therefore yielding inferior CNN-LSTM configurations in addressing complicated trends, e.g. fluctuation and volatility, in energy forecasting tasks. In short, in comparison with the baseline methods, the proposed diverse search strategies, e.g. the nonlinear exploration rate adjustment, the chaotic leadership rivalries, as well as Lévy random jumps, account for the superior performance of the proposed evolving CNN-LSTM network.

B. PM2.5 CONCENTRATION PREDICTION

1) DATA SET

To further assess model efficiency, we employ the UCI Beijing air quality data set2[2] [88] for PM2.5 concentration prediction using the devised evolving CNN-LSTM networks. This data set includes hourly measurements of four types of air pollutants, i.e. SO2, NO2, CO, and O3, as well as five meteorological parameters, i.e. temperature, pressure, dew point temperature, amount of precipitation, and wind speed, over a four-year period from 1 March 2013 to 28 February 2017. A reliable prediction of PM2.5 concentrations requires an accurate interpretation of the changing patterns of air pollutants under various temporal contexts, which poses a great challenge to the devised CNN-LSTM networks.

2) EXPERIMENTAL SETTINGS

Similar to the method used the energy forecasting task, a multi-input and multi-output time series model is established to predict the PM2.5 concentrations in the air in Beijing for a week in advance, based on the historical data from the previous two weeks. The hourly recordings are transformed into daily measurements to better understand weekly periodicity of input variables as well as to make weekly prediction of PM2.5 concentrations. The vector of the input sequence is 14 × 9, where 14 and 9 represent time steps and the feature size, respectively. The same experimental setting as that in the previous energy consumption problem is used, since both are time-series forecasting tasks. Besides that, the data samples from the first and second years are used for training, while those from the third and last years are used for validation and test, respectively.

3) RESULTS AND DISCUSSION

As shown in TABLE 6 and TABLE 7, the optimized CNN-LSTM networks identified by the proposed GWO algorithm yield more robust and reliable predictions for weekly PM2.5 concentrations in comparison with those of the seven baseline methods and the default CNN-LSTM network. The devised CNN-LSTM networks achieve the smallest average results of RMSE and MAE, i.e. 62.2 and 40.8, over ten independent runs, whereas the baseline methods, in general, produce less favorable results with high variances and inconsistencies across ten different runs. In particular, the RMSE scores are reduced by 6.2%, 13.1%, 9.1%, and 15.1%, by the devised CNN-LSTM networks, in comparison with those from GWO, prLeGWO, FuzzyGWO, and the default CNN-LSTM model, respectively. The significant performance improvements of the devised CNN-LSTM networks can be further observed from the MAE results. The superiority in performance indicates the effectiveness of

TABLE 6. The RMSE results over 10 independent runs.

Run	CNN-LSTM	GWO	PSO	GSA	FPA	CSO	prLe GWO	Fuzzy GWO	Prop. GWO
1	67.3	64.7	64.3	70.0	64.2	92.1	82.4	68.3	62.6
2	70.8	65.2	65.6	62.6	67.7	64.7	61.0	70.1	63.5
3	68.6	62.0	63.9	65.8	78.9	66.1	65.1	60.5	62.8
4	74.6	65.9	64.9	64.8	58.1	63.9	64.7	63.7	65.6
5	75.4	70.6	64.9	73.5	65.9	67.8	66.1	67.3	59.9
6	74.2	65.8	63.8	69.1	73.2	64.7	117.3	72.4	62.3
7	69.3	67.7	61.0	63.6	76.7	60.0	62.3	72.7	61.6
8	73.7	63.2	64.7	68.7	70.7	68.3	61.6	67.6	59.9
9	87.5	73.0	61.7	63.4	69.2	68.1	71.7	69.4	62.5
10	71.6	65.4	68.1	64.0	77.2	69.0	63.4	72.4	61.3
Avg.	73.3	66.3	64.3	66.6	70.2	68.5	71.6	68.4	**62.2**

TABLE 7. The MAE results over 10 independent runs.

Run	CNN-LSTM	GWO	PSO	GSA	FPA	CSO	prLe GWO	Fuzzy GWO	Prop. GWO
1	44.8	42.6	41.4	46.2	40.2	54.7	55.2	45.1	39.6
2	47.6	41.6	43.9	42.2	44.6	45.7	41.9	49.1	42.5
3	48.6	41.4	41.6	43.9	52.3	43.5	42.9	42.9	41.1
4	49.8	43.1	42.0	43.0	40.8	40.6	42.4	40.8	43.6
5	51.1	44.7	42.6	48.5	43.5	44.7	42.4	44.2	40.3
6	49.9	43.0	41.8	43.6	48.2	43.0	70.3	52.5	40.4
7	45.1	41.6	39.4	42.4	53.0	39.7	42.8	46.5	39.8
8	48.5	42.0	44.2	44.2	46.8	44.9	41.4	46.3	39.4
9	54.3	46.7	40.4	41.7	46.0	45.6	47.3	46.3	40.9
10	48.9	42.8	42.6	42.1	53.7	45.7	42.5	52.6	40.3
Avg.	48.9	42.9	42.0	43.8	46.9	44.8	46.9	46.6	**40.8**

[2] The URL of the Beijing air quality data set is: https://archive.ics.uci.edu/ml/datasets/Beijing+Multi-Site+Air-Quality+Data.

the proposed evolving model in extracting effective features and recognizing complex temporal variations embedded in time-series air pollution data as well as in dynamic meteorological conditions.

Moreover, the mean hyperparameters of the identified optimal structures for PM2.5 concentration predictions over ten independent runs are presented in TABLE 8. The main characteristics of the effective CNN-LSTM configurations in the PM2.5 prediction are similar to those demonstrated in energy forecasting. The optimized CNN-LSTM structures produced by the proposed GWO variant possess a relatively larger number of filters in the first convolutional layer, i.e. 132.8, while maintaining smaller numbers of nodes in both the LSTM and dense layers, i.e. 126.2 and 51.4, respectively. Such compositions enable an efficient extraction of the most important features among meteorological variables and air pollutants in the convolutional layers, while endowing the optimized CNN-LSTM networks with sufficient representational capacities to effectively capture various dependencies in the LSTM and dense layers, in order to avoid overfitting.

TABLE 8. **The mean configurations of the identified cnn-lstm networks over 10 runs.**

Conf.	GWO	PSO	GSA	FPA	CSO	prLe GWO	Fuzzy GWO	Prop. GWO
No. 1st	35.2	165.6	198.4	39.6	72.8	3.8	4.6	132.8
No. 2st	2	2	26.8	118.4	14.8	2.6	2.6	2.4
S. 1st	1	1.7	2.9	1.8	2.2	2.5	1.4	1.2
S. 2st	1.3	1.8	3.1	2.6	1.9	2.4	1.4	1.3
LSTM	85.2	237.2	271.2	200.5	223.7	124.9	81.2	126.2
Dense	60	53.9	91.3	95.3	76.8	85.3	15.9	51.4
DR	0.305	0.352	0.321	0.219	0.370	0.018	0.004	0.313
LR	0.034	0.048	0.049	0.028	0.058	0.007	0.001	0.046

To be specific, the employed air pollution data set not only contains important factors in relation to the generation and dispersion of PM2.5 concentrations, e.g. SO2, NO2, and wind speed, but also disturbing factors with various confounding effects, e.g. CO and O3. Therefore, the prediction of PM2.5 is a challenging and complex task. As such, a proper feature extraction capability is required to identify the discriminative features that represent the complex formation mechanism of PM2.5, as well as sophisticated aerodynamic effects on its dilution. The RMSE and MAE results indicate that our optimized CNN-LSTM networks are able to resolve the challenging factors more effectively and demonstrate greater resilience in handling temporal variances and interactions among variables. In other words, the identified filter structures in the convolutional layers are capable of generating informative feature maps, which can both uncover the indirect impacts of various pollutants permeated in the air, as well as the direct impacts of weather conditions, on the concentration of PM2.5. Meanwhile, the identified optimized configurations of the LSTM and dense layers are able to better comprehend and capture the long-term dependencies among the input data sequences. As such, the devised CNN-LSTM structures identified by the proposed GWO variant are proven to be superior in undertaking complex PM2.5 concentration prediction tasks.

C. HUMAN ACTIVITY RECOGNITION

1) DATA SET

In addition to time-series prediction, we use a time series classification task for evaluation of the CNN-LSTM networks, i.e. the UCI human activity recognition (HAR) data set[3] [89]. The data set was collected from 30 volunteers performing six types of daily living activities, i.e. standing, sitting, laying down, walking, walking downstairs and upstairs, while carrying a waist-mounted smartphone embedded with inertial sensors. Three types of signals, including total acceleration, body acceleration, and body gyroscope, were recorded with a sampling rate of 50Hz. These sensor signals were pre-processed using noise filters and sampled with a sliding window of 2.56 sec, i.e. 128 readings, with a 50% overlap. The input vector is therefore 128×9, in which 128 and 9 denote the number of readings and the number of features, respectively. The total sample sizes in the training and test data sets are 7,352 and 2,947, respectively.

2) EXPERIMENTAL SETTINGS

In this HAR task, the nine hyperparameters in relation to network capacities and learning properties listed in TABLE 1 are optimized. The training process is divided into two main stages. Firstly, the optimal configuration of CNN-LSTM is identified by the proposed GWO variant using a smaller proportion of the training data, in order to reduce the computational load. Specifically, the first 3000 samples in the training data set are used for training, and the subsequent 1500 samples for validation, in order to the search for the optimal network configuration. In the training process, the Adam optimizer is adopted, while the categorical cross-entropy is applied as the loss function. The batch size and epoch number are set to 256 and 20, respectively. Besides that, the error rate is employed as the fitness score to be minimized during the evolving process. Subsequently, the CNN-LSTM model with the identified optimal structure is retrained for 100 epochs using the whole training data set of 7,352 samples. The obtained CNN-LSTM model is then used to perform classification of human activities based on the unseen test data set with 2,947 samples.

3) RESULTS AND DISCUSSION

A total of four performance indicators are employed to evaluate the effectiveness of the optimized CNN-LSTM networks in classifying the human activities, i.e. accuracy, F-score, precision, and recall. The overall results over ten independent runs are presented in TABLE 9 to TABLE 12.

With respect to accuracy, the CNN-LSTM configurations yielded by the proposed GWO variant achieve the highest mean accuracy rate of 92.3%, outperforming those identified

[3] The URL of the human activity recognition data set is: https://archive.ics.uci.edu/ml/datasets/human+activity+recognition+using+smartphones.

TABLE 9. The results of classification accuracy over 10 independent runs.

Run	CNN-LSTM	GWO	PSO	GSA	FPA	CSO	prLe GWO	Fuzzy GWO	Prop. GWO
1	0.879	0.886	0.907	0.911	0.881	0.893	0.863	0.883	0.928
2	0.882	0.889	0.900	0.904	0.815	0.907	0.886	0.928	0.929
3	0.859	0.855	0.909	0.898	0.888	0.879	0.883	0.900	0.922
4	0.879	0.862	0.897	0.910	0.882	0.894	0.864	0.880	0.916
5	0.877	0.876	0.916	0.904	0.909	0.892	0.856	0.902	0.921
6	0.872	0.889	0.909	0.904	0.895	0.889	0.870	0.917	0.931
7	0.880	0.898	0.918	0.883	0.899	0.877	0.879	0.882	0.918
8	0.888	0.891	0.891	0.901	0.906	0.899	0.845	0.859	0.929
9	0.885	0.900	0.900	0.896	0.890	0.923	0.796	0.864	0.914
10	0.863	0.902	0.883	0.903	0.890	0.909	0.877	0.880	0.922
Avg.	0.877	0.885	0.903	0.901	0.886	0.896	0.862	0.889	**0.923**

TABLE 10. The results of F-score over 10 independent runs.

Run	CNN-LSTM	GWO	PSO	GSA	FPA	CSO	prLe GWO	Fuzzy GWO	Prop. GWO
1	0.880	0.886	0.907	0.912	0.881	0.892	0.863	0.883	0.928
2	0.884	0.891	0.900	0.905	0.813	0.908	0.886	0.930	0.930
3	0.857	0.854	0.910	0.898	0.888	0.878	0.883	0.901	0.925
4	0.879	0.861	0.896	0.909	0.882	0.894	0.860	0.880	0.916
5	0.876	0.876	0.916	0.903	0.908	0.892	0.856	0.903	0.920
6	0.871	0.888	0.909	0.905	0.895	0.891	0.870	0.917	0.924
7	0.880	0.898	0.917	0.883	0.898	0.876	0.880	0.880	0.918
8	0.887	0.890	0.890	0.900	0.906	0.899	0.843	0.858	0.931
9	0.884	0.900	0.899	0.896	0.890	0.922	0.792	0.861	0.914
10	0.861	0.902	0.884	0.902	0.889	0.909	0.874	0.879	0.925
Avg.	0.876	0.885	0.903	0.901	0.885	0.896	0.861	0.889	**0.923**

TABLE 11. The results of precision over 10 independent runs.

Run	CNN-LSTM	GWO	PSO	GSA	FPA	CSO	prLe GWO	Fuzzy GWO	Prop. GWO
1	0.883	0.887	0.909	0.915	0.887	0.892	0.872	0.891	0.930
2	0.885	0.893	0.901	0.908	0.818	0.907	0.889	0.932	0.931
3	0.859	0.857	0.912	0.899	0.894	0.878	0.886	0.904	0.927
4	0.878	0.861	0.899	0.908	0.890	0.896	0.862	0.883	0.917
5	0.876	0.878	0.917	0.909	0.908	0.894	0.861	0.908	0.921
6	0.871	0.893	0.911	0.909	0.898	0.891	0.871	0.917	0.925
7	0.880	0.900	0.917	0.883	0.899	0.877	0.884	0.881	0.920
8	0.887	0.895	0.889	0.901	0.907	0.900	0.854	0.859	0.933
9	0.884	0.901	0.898	0.896	0.890	0.922	0.796	0.868	0.915
10	0.863	0.905	0.884	0.903	0.892	0.910	0.874	0.884	0.927
Avg.	0.877	0.887	0.904	0.903	0.888	0.897	0.865	0.893	**0.925**

TABLE 12. The results of Recall over 10 independent runs.

Run	CNN-LSTM	GWO	PSO	GSA	FPA	CSO	prLe GWO	Fuzzy GWO	Prop. GWO
1	0.881	0.887	0.909	0.913	0.881	0.894	0.861	0.883	0.928
2	0.885	0.891	0.900	0.905	0.816	0.910	0.888	0.930	0.931
3	0.857	0.853	0.910	0.900	0.888	0.878	0.882	0.902	0.924
4	0.880	0.863	0.899	0.911	0.882	0.895	0.860	0.882	0.918
5	0.878	0.879	0.916	0.905	0.908	0.894	0.854	0.904	0.922
6	0.872	0.886	0.910	0.905	0.896	0.892	0.873	0.919	0.924
7	0.881	0.901	0.917	0.885	0.899	0.876	0.881	0.879	0.918
8	0.889	0.891	0.892	0.903	0.907	0.899	0.841	0.861	0.931
9	0.885	0.902	0.901	0.898	0.893	0.923	0.789	0.861	0.916
10	0.864	0.902	0.885	0.903	0.891	0.910	0.875	0.877	0.924
Avg.	0.877	0.885	0.904	0.903	0.886	0.897	0.860	0.890	**0.924**

by all baseline models. In particular, the proposed GWO variant demonstrates significant advantages than the original GWO and advanced GWO variants, i.e. prLeGWO and FuzzyGWO, and the default CNN-LSTM network, with performance difference of 3.8%, 6.1%, 3.4%, and 4.6%, respectively. In addition, similar superiorities of the proposed GWO model can be observed consistently across the remaining indicators, i.e. F-score, precision, and recall scores, as shown in TABLE 10 to TABLE 12.

The decomposed accuracy results with respect to each of the six human activities are provided in TABLE 13. The optimized CNN-LSTM networks yielded by the proposed GWO variant produce the highest accuracy results on four activity classes, i.e. walking, walking upstairs, walking downstairs, and standing, significantly outperforming the baseline methods and default network with evident performance gaps. This indicates that the CNN-LSTM configurations yielded by the proposed GWO variant can successfully discover distinctive variations and discriminative patterns with respect to different human activities, therefore achieving better performances. In other words, the decomposed results further reinforce the effectiveness and superiority of the proposed GWO variant in identifying the most effective deep networks for undertaking this HAR task, in comparison with the baseline models.

TABLE 13. The mean accuracy rate of each class over 10 independent runs.

Class	CNN-LSTM	GWO	PSO	GSA	FPA	CSO	prLe GWO	Fuzzy GWO	Prop. GWO
Walk	0.905	0.915	0.935	0.924	0.908	0.935	0.886	0.941	**0.973**
W-Up	0.890	0.917	0.905	0.908	0.878	0.907	0.843	0.906	**0.942**
W-Dn	0.918	0.923	0.956	0.980	0.944	0.951	0.863	0.912	**0.984**
Sit	0.787	0.768	**0.808**	0.778	0.773	0.792	0.768	0.785	0.791
Stand	0.787	0.829	0.835	0.863	0.837	0.825	0.838	0.847	**0.891**
Lay	0.976	0.961	**0.984**	0.966	0.976	0.974	0.966	0.966	0.964

Moreover, the mean hyperparameters of the optimized CNN-LSTM networks over ten independent runs are presented in TABLE 14. In particular, the devised CNN-LSTM networks possess the highest numbers of filters in both convolutional layers, i.e. 230.8 and 193.6, respectively, while maintaining fewer numbers of nodes in the recurrent and dense layers, i.e. 60.2 and 41.2, respectively, in comparison with those of the baseline models and the default network settings. Such configurations enable CNN-LSTM

TABLE 14. The mean configurations of the identified cnn-lstm networks over 10 runs.

Conf.	GWO	PSO	GSA	FPA	CSO	prLe GWO	Fuzzy GWO	Prop. GWO
No. 1st	38.8	94.4	51.2	64.2	125.4	68.5	59.2	230.8
No. 2st	59.0	80.2	60.0	18.8	57.2	130.3	92.6	193.6
S. 1st	4.0	3.7	3.7	3.7	4.4	3.6	3.6	4.4
S. 2st	4.5	3.3	3.7	3.6	3.5	3.3	3.6	3.6
Pool.	3.4	3.4	3.6	3.3	3.8	3.3	3.6	2.7
LSTM	68.5	102.1	102.1	126.3	97.7	34.1	100.3	60.2
Dense	92.9	104.3	110.7	105.8	131.2	20.4	108.9	41.2
DR	0.376	0.427	0.293	0.214	0.195	0.143	0.246	0.416
LR	0.037	0.059	0.046	0.046	0.048	0.020	0.028	0.029

to thoroughly examine the fundamental characteristics with respect to each category of human activity and differentiate subtle differences between them. As a result, the most discriminative features related to human activities can be extracted by the convolutional layers, while achieving efficient trade-offs between learning long-term dependencies embedded among the consecutive body movements and avoiding overfitting on noise in the recurrent and dense layers. As such, the CNN-LSTM networks identified by the proposed GWO variant are able to distinguish different human activities effectively.

D. REMARKS

Overall, the proposed GWO variant is capable of identifying the most effective CNN-LSTM configurations with appropriate representational capacities and superior capabilities of feature extraction, for resolving all three employed time series tasks. In contrast, the baseline search methods yield less effective sub-optimal CNN-LSTM networks with oversized or undersized hyperparameters, which result in performance degradation. Specifically, the oversized settings in the recurrent and dense layers and the lack of regularization are likely to cause overfitting owing to the excessive representational capacities and memorizing of sample noise, as indicated by the results of GWO and CSO on energy consumption forecasting, FPA and CSO on PM2.5 concentration prediction, as well as FPA and FuzzyGWO on HAR. Moreover, the undersized network configurations produce oversimplified CNN-LSTM structures with restricted interpretation capabilities, therefore unable to fully capture sophisticated dependencies embedded in variables under complex temporal contexts, neither to perform effective feature extraction and transformation, as exemplified by the results of FuzzyGWO on PM2.5 prediction, and prLeGWO on HAR. Furthermore, our optimized networks also outperform the CNN-LSTM model with the default hyperparameter settings significantly in all three experiments, owing to the limitations of the pre-determined inefficient model and training configurations in such default networks, i.e. the lack of learnable filters for feature extraction and memorizing of sample noise resulted from the redundant recurrent memory cells. To sum up, the proposed GWO variant demonstrates significant advantages over the baseline models in automatic identification of the optimal CNN-LSTM configurations for undertaking all three time series tasks, owing to the enhanced search diversity and search efficiency.

In terms of computational time, the proposed model takes 7-12 hours for hyperparameter fine-tuning for each run in our experimental studies. The mean computational cost of one fitness evaluation with respect to each search method is presented in TABLE 15. This is the average cost of training and evaluation of an optimized model in one fitness evaluation using the training and validation sets, respectively. TABLE 15 provides an indication on the computational times from different search methods. On average, the proposed GWO variant requires a lower computational cost for each fitness evaluation as compared with those from the majority of the baseline methods in all three time series prediction tasks. The computational efficiency of our devised CNN-LSTM models can be attributed to the characteristics of the identified network configurations, i.e. smaller numbers of nodes in both the LSTM and fully connected dense layers. As a result, the attenuated connections of the fully connected layer as well as the lighter settings in the LSTM layer can reduce the network complexity, therefore lowering the computational costs.

TABLE 15. The mean computational cost of each fitness evaluation for each search method (in seconds).

	GWO	PSO	GSA	FPA	CSO	prLe GWO	Fuzzy GWO	Prop. GWO
Energy	13.13	14.69	11.31	13.76	8.47	7.55	6.88	6.96
PM2.5	16.88	17.47	20.32	25.61	17.12	16.13	14.32	14.85
HAR	6.95	9.24	9.18	10.31	7.77	6.63	10.51	6.71

E. WILCOXON STATISTICAL TEST

The Wilcoxon statistical rank sum test is conducted to further indicate the statistical distinctiveness of the enhanced GWO model against the baseline methods in searching for the optimal CNN-LSTM configurations. The accuracy results of HAR and RMSE results of energy consumption forecast and PM2.5 concentration prediction are employed for the statistical analysis. As shown in TABLE 16, the rank sum test results are lower than 0.05, which indicate that the proposed GWO variant significantly outperforms all the baseline search methods from the statistical perspective, including four classical methods, i.e. GWO, PSO, GSA, and FPA, and three advanced variant models, i.e. CSO, prLeGWO, and FuzzyGWO, in identifying the optimal CNN-LSTM configurations for undertaking time series prediction and classification problems. Our devised optimized networks also show statistically significant superiority over those with default settings. This superiority of the proposed GWO variant can be attributed to the improved trade-offs between search diversification and intensification facilitated by the cooperation among the proposed comprehensive and complementary search strategies. A detailed analysis is provided, as follows.

TABLE 16. Wilcoxon rank sum test results over 10 independent runs.

	CNN-LSTM	GWO	PSO	GSA	FPA	CSO	prLe GWO	Fuzzy GWO
HAR	1.80E-04	1.81E-04	4.35E-04	1.81E-04	1.81E-04	9.99E-04	1.81E-04	2.20E-03
Energy	1.83E-04	3.61E-03	2.57E-02	2.20E-03	3.61E-03	3.61E-03	4.52E-02	3.61E-03
PM2.5	1.83E-04	3.30E-04	7.69E-04	2.46E-04	1.13E-02	3.61E-03	1.13E-02	3.30E-04

The key challenge in searching for effective CNN-LSTM configurations lies in the complex interactions among different components within the network, as well as high computational costs. In this regard, the proposed GWO variant incorporates several distinctive and complementary strategies, capable of boosting search diversity and improving the convergence speed to resolve the challenges occurred during the exploration of the optimal CNN-LSTM configurations. Specifically, an advanced trade-off between search diversification and intensification is achieved by the proposed nonlinear adjustment of the territory boundary. Under this scheme, the search range during the exploration is maintained at the initial level without any acute decrease, enabling the wolf population to conduct extensive explorations around the peripheral areas of the search territory, instead of being drawn to the vicinity of the leading wolves at the beginning of the search process. Meanwhile, this transition scheme also enables the wolf population to focus on the closer bounds around the leading wolves and conduct thorough detection surrounding the promising regions during exploitation. In addition, the proposed sinusoidal chaotic leadership rivalry enables the GWO variant to leverage the merits from both multiple-leader guided search and single-leader guided search, through reinforcing the leadership of the best wolf solution while periodically downplaying the influence of the global best solution in position updating. As such, a periodic balance between search diversity and concentration is achieved. Thirdly, the fine-tuning capability around the global best position is improved by conducting refined local detections with various steps and directions at the final stage of the search process, using a dedicated damped function with a dynamic adjustment of the amplitude. Lastly, the qualities of three leading wolves are further enhanced using the Lévy flight probability distributions to reduce the likelihood of stagnation at local optima.

Overall, the effectiveness of the proposed GWO variant can be ascribed to the enhanced search diversity and search efficiency. The diversity is improved from three perspectives, i.e. the upholding of the search territory boundary through the dedicated nonlinear control of the exploration factor, the diversification of leading signals by the chaotic allocation of leadership weights, as well as leader random walks based on Lévy flight. Meanwhile, the efficiency is achieved from two perspectives, i.e. the ascertained dominance of the best wolf leader during the search process, and the dedicated local exploitation around the global best solution at the final stage of the search process. As such, the enhanced GWO model is more likely to escape from local stagnation and attain the global optimality. Therefore, the complex interactions among CNN-LSTM hyperparameters can be thoroughly explored by the proposed GWO variant, and effective CNN-LSTM configurations can be identified swiftly. The efficiency of the proposed GWO-based CNN-LSTM network is evidenced by the superior empirical results on the three time series problems, supported by the statistical test results. In contrast, the baseline GWO variants, e.g. prLeGWO and FuzzyGWO, achieve less efficient trade-offs between reassuring the dominance of the best leader and retaining diversity in reconstruction of leadership hierarchy. Besides that, there is a lack of refinement in terms of the transition between exploration and exploitation among the baseline GWO variants and other search methods. Overall, the enhanced GWO algorithm demonstrates great advantages in devising optimal CNN-LSTM networks and outperforms eight baseline methods significantly in undertaking time series prediction and classification tasks.

V. CONCLUSION AND FUTURE DIRECTIONS

We have proposed an evolving CNN-LSTM network to solve time series prediction and classification problems. A GWO variant has been proposed for automatic optimal hyperparameter and topology identification of the network architectures. The proposed GWO variant employs a nonlinear exploration rate for search boundary adjustment, a sinusoidal chaotic map for the leadership allocation pertaining to the dominant wolf leaders, an enhanced spiral local exploitation scheme, as well as a Lévy flight-based leader enhancement mechanism. As such, the search process becomes more diversified owing to the expansion of the search territory, random exploitation of the wolf leaders, and chaotic aggregation and periodical diversification of the guiding signals. In addition, the search efficiency and convergence rate are improved owing to the dominance of the global best wolf leader over the combined distractions from the remaining two leaders during the search process, as well as the intensified local exploitation around the global best solution at the final search stage.

The proposed GWO-based evolving CNN-LSTM model has been evaluated using two time series prediction problems, i.e. energy consumption forecast and PM2.5 concentration prediction, and a time series classification task, i.e. HAR. Our devised evolving deep networks outperform the default network and those yielded by a total of seven baseline search models including four classical search methods and three advanced GWO and PSO variants on all the test data sets, with statistically significant difference in performance. Moreover, the empirical results indicate that our optimized CNN-LSTM networks are characterized by a higher number of filters in the convolutional layers and moderate settings in terms of the numbers of nodes in the LSTM layer and the fully connected layer. Such devised networks possess superior capabilities in capturing temporal and sequential information over those identified by all the baseline methods for undertaking time-series prediction and classification tasks. In other words, the identified optimal network configurations are able to thoroughly examine the interactions among time series variables, and provide efficient network representational capacities without suffering from either overfitting or underfitting issues.

For future research, we aim to deploy the proposed GWO-optimized evolving CNN-LSTM model for tackling other sophisticated time series prediction tasks, such as EEG-based medical diagnosis [90], video classification [91], and language generation [92]. Besides that, investigations on deep architecture generation with residual and dense connectivity using the proposed GWO algorithm for large-scale object detection and recognition [93], image segmentation [94], [95], and image description generation [96] problems will be conducted.

From the algorithmic perspective, in addition to greedy search, we will investigate other potential effective interaction schemes between the wolf leaders and the remaining wolf individuals. The aim is to maximize the adaptation of the wolf population in the long run, i.e. sacrificing short-term rewards for long-term benefits during evolution. On the other hand, we will explore advanced local search strategies by incorporating personal best experiences [42] or merits from other metaheuristic algorithms [97], [98], to further enhance the proposed GWO algorithm. Other LSTM variants will also be explored to increase efficiency of the resulting network.

REFERENCES

[1] A. S. Weigend, *Time Series Prediction: Forecasting the Future and Understanding the Past*. Westview Press, Santa Fe Institute Studies in the Sciences of Complexity, Nov. 1993.

[2] C.-Y. Zhang, C. L. P. Chen, M. Gan, and L. Chen, "Predictive deep Boltzmann machine for multiperiod wind speed forecasting," *IEEE Trans. Sustain. Energy*, vol. 6, no. 4, pp. 1416–1425, Oct. 2015.

[3] D. T. Tran, A. Iosifidis, J. Kanniainen, and M. Gabbouj, "Temporal attention-augmented bilinear network for financial time-series data analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 5, pp. 1407–1418, May 2019.

[4] G. Valenza, M. Nardelli, A. Lanata, C. Gentili, G. Bertschy, M. Kosel, and E. P. Scilingo, "Predicting mood changes in bipolar disorder through heartbeat nonlinear dynamics," *IEEE J. Biomed. Health Informat.*, vol. 20, no. 4, pp. 1034–1043, Jul. 2016.

[5] J. Liu, C. Wang, and Y. Liu, "A novel method for temporal action localization and recognition in untrimmed video based on time series segmentation," *IEEE Access*, vol. 7, pp. 135204–135209, 2019.

[6] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*. Hoboken, NJ, USA: Wiley, 2015.

[7] G. E. P. Box and D. A. Pierce, "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," *J. Amer. Stat. Assoc.*, vol. 65, no. 332, pp. 1509–1526, Dec. 1970.

[8] H. Drucker, C. Burges, L. Kaufman, A. J. Smola, and V. Vapnik, "Support vector regression machines," in *Proc. NIPS*, 1996, pp. 155–161.

[9] M. Khashei and M. Bijari, "An artificial neural network (p,d,q) model for timeseries forecasting," *Expert Syst. Appl.*, vol. 37, no. 1, pp. 479–489, Jan. 2010.

[10] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 240–254, Mar. 1994.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, pp. 1735–1780, 1997.

[12] S. F. Crone and N. Kourentzes, "Feature selection for time series prediction—A combined filter and wrapper approach for neural networks," *Neurocomputing*, vol. 73, nos. 10–12, pp. 1923–1936, Jun. 2010.

[13] C. Wong and M. Versace, "CARTMAP: A neural network method for automated feature selection in financial time series forecasting," *Neural Comput. Appl.*, vol. 21, no. 5, pp. 969–977, Jul. 2012.

[14] B. Nakisa, M. N. Rastgoo, A. Rakotonirainy, F. Maire, and V. Chandran, "Long short term memory hyperparameter optimization for a neural network based emotion recognition framework," *IEEE Access*, vol. 6, pp. 49325–49338, 2018.

[15] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.

[16] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[17] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: A review of recent variants and applications," *Neural Comput. Appl.*, vol. 30, no. 2, pp. 413–435, Jul. 2018.

[18] Y. Zhang, X. Liu, F. Bao, J. Chi, C. Zhang, and P. Liu, "Particle swarm optimization with adaptive learning strategy," *Knowl.-Based Syst.*, vol. 196, May 2020, Art. no. 105789.

[19] N. Lynn and P. N. Suganthan, "Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation," *Swarm Evol. Comput.*, vol. 24, pp. 11–24, Oct. 2015.

[20] R. Hassan, B. Cohanim, O. Weck, and G. Venter, "A comparison of particle swarm optimization and the genetic algorithm," in *Proc. 46th AIAA/ASME/ASCE/AHS/ASC-Conf. Struct., Dyn. Mater.*, 2005, pp. 1–13.

[21] J. Vesterstrom and R. Thomsen, "A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems," in *Proc. Congr. Evol. Comput. (CEC)*, 2004, pp. 1980–1987.

[22] X. Song, L. Tang, S. Zhao, X. Zhang, L. Li, J. Huang, and W. Cai, "Grey wolf optimizer for parameter estimation in surface waves," *Soil Dyn. Earthq. Eng.*, vol. 75, pp. 147–157, Aug. 2015.

[23] E. Gupta and A. Saxena, "Grey Wolf optimizer based regulator design for automatic generation control of interconnected power system," *Cogent Eng.*, vol. 3, no. 1, Mar. 2016, Art. no. 1151612.

[24] S. Ardabili, A. Mosavi, P. Ghamisi, F. Ferdinand, A. Koczy, U. Reuter, T. Rabczuk, and P. Atkinson, "COVID-19 outbreak prediction with machine learning," *medRxiv*, to be published. [Online]. Available: https://ssrn.com/abstract=3580188 and http://dx.doi.org/10.2139/ssrn.3580188

[25] S. Li and J. Wang, "Dynamic modeling of steam condenser and design of pi controller based on Grey wolf optimizer," *Math. Probl. Eng.*, vol. 2015, Dec. 2015, Art. no. 120975.

[26] V. K. Kamboj, S. K. Bath, and J. S. Dhillon, "Solution of non-convex economic load dispatch problem using Grey wolf optimizer," *Neural Comput. Appl.*, vol. 27, no. 5, pp. 1301–1316, Jul. 2016.

[27] S. Mirjalili, "How effective is the Grey wolf optimizer in training multi-layer perceptrons," *Appl. Intell.*, vol. 43, no. 1, pp. 150–161, 2015.

[28] S. A. Medjahed, T. A. Saadi, A. Benyettou, and M. Ouali, "Gray Wolf optimizer for hyperspectral band selection," *Appl. Soft Comput.*, vol. 40, pp. 178–186, Mar. 2016.

[29] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary Grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, Jan. 2016.

[30] A. K. M. Khairuzzaman and S. Chaudhury, "Multilevel thresholding using Grey wolf optimizer for image segmentation," *Expert Syst. Appl.*, vol. 86, pp. 64–76, Nov. 2017.

[31] K. Li, G. Zhou, Y. Yang, F. Li, and Z. Jiao, "A novel prediction method for favorable reservoir of oil field based on Grey wolf optimizer and twin support vector machine," *J. Petroleum Sci. Eng.*, vol. 189, Jun. 2020, Art. no. 106952.

[32] S. Zhang, Y. Zhou, Z. Li, and W. Pan, "Grey wolf optimizer for unmanned combat aerial vehicle path planning," *Adv. Eng. Softw.*, vol. 99, pp. 121–136, Sep. 2016.

[33] G. M. Komaki and V. Kayvanfar, "Grey wolf optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time," *J. Comput. Sci.*, vol. 8, pp. 109–120, May 2015.

[34] J.-S. Wang and S.-X. Li, "An improved Grey wolf optimizer based on differential evolution and elimination mechanism," *Sci. Rep.*, vol. 9, no. 1, p. 7181, Dec. 2019.

[35] P. Hu, S. Chen, H. Huang, G. Zhang, and L. Liu, "Improved alpha-guided Grey wolf optimizer," *IEEE Access*, vol. 7, pp. 5421–5437, 2019.

[36] A. A. Heidari and P. Pahlavani, "An efficient modified Grey wolf optimizer with Lévy flight for optimization tasks," *Appl. Soft Comput.*, vol. 60, pp. 115–134, Nov. 2017.

[37] F. B. Ozsoydan, "Effects of dominant wolves in Grey wolf optimization algorithm," *Appl. Soft Comput.*, vol. 83, Oct. 2019, Art. no. 105658.

[38] K. Luo, "Enhanced Grey wolf optimizer with a model for dynamically estimating the location of the prey," *Appl. Soft Comput.*, vol. 77, pp. 225–235, Apr. 2019.

[39] S. Gupta and K. Deep, "A novel random walk Grey wolf optimizer," *Swarm Evol. Comput.*, vol. 44, pp. 101–112, Feb. 2019.

[40] E. Emary, H. M. Zawbaa, and C. Grosan, "Experienced gray wolf optimization through reinforcement learning and neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 681–694, Mar. 2018.

[41] Q. Tu, X. Chen, and X. Liu, "Hierarchy strengthened Grey wolf optimizer for numerical optimization and feature selection," *IEEE Access*, vol. 7, pp. 78012–78028, 2019.

[42] S. Gupta and K. Deep, "A memory-based Grey wolf optimizer for global optimization tasks," *Appl. Soft Comput.*, vol. 93, Aug. 2020, Art. no. 106367.

[43] R. A. Ibrahim, M. A. Elaziz, and S. Lu, "Chaotic opposition-based Grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization," *Expert Syst. Appl.*, vol. 108, pp. 1–27, Oct. 2018.

[44] M. A. Al-Betar, M. A. Awadallah, H. Faris, I. Aljarah, and A. I. Hammouri, "Natural selection methods for Grey wolf optimizer," *Expert Syst. Appl.*, vol. 113, pp. 481–498, Dec. 2018.

[45] W. Long, J. Jiao, X. Liang, and M. Tang, "Inspired Grey wolf optimizer for solving large-scale function optimization problems," *Appl. Math. Model*, vol. 60, pp. 112–126, 2018.

[46] A. Saxena, R. Kumar, and S. Das, "ß-chaotic map enabled Grey wolf optimizer," *Appl. Soft Comput.*, vol. 75, pp. 84–105, Feb. 2019.

[47] Z. Beheshti, and S.M. Shamsuddin, "Future paths for integer programming and links to artificia intelligence," *Comput. Oper. Res.*, vol. 13, no. 5, pp. 533–549, Jan. 1986.

[48] D. Wolpert and W. Macready, "No free lunch theorems for search," Santa Fe Inst., Santa Fe, NM, USA, Tech. Rep. SFI-TR-95-02-010, 1995.

[49] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Boston, MA, USA: Addison-Wesley, 1989.

[50] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[51] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw. (ICNN)*, 1995, pp. 1942–1948.

[52] H. Xie, L. Zhang, C. P. Lim, Y. Yu, C. Liu, H. Liu, and J. Walters, "Improving K-means clustering with enhanced firefly algorithms," *Appl. Soft Comput.*, vol. 84, Nov. 2019, Art. no. 105763.

[53] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.

[54] E. Rashedi, H. Nezamabadi-pour, and S. Saryazdi, "GSA: A gravitational search algorithm," *Inf. Sci.*, vol. 179, no. 13, pp. 2232–2248, Jun. 2009.

[55] X. S. Yang, "Flower pollination algorithm for global optimization," in *Unconventional Computation and Natural Computation*. Berlin, Germany: Springer, 2012.

[56] S. Li, H. Chen, M. Wang, A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Gener. Comp. Syst.*, vol. 111, pp. 300–323, Oct. 2020.

[57] Q. Askari, M. Saeed, and I. Younas, "Heap-based optimizer inspired by corporate rank hierarchy for global optimization," *Expert Syst. Appl.*, vol. 161, Dec. 2020, Art. no. 113702.

[58] I. Ahmadianfar, O. Bozorg-Haddad, and X. Chu, "Gradient-based optimizer: A new Metaheuristic optimization algorithm," *Inf. Sci.*, vol. 540, pp. 131–159, Nov. 2020.

[59] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.

[60] Y. Sun, B. Xue, M. Zhang, G. G. Yen, and J. Lv, "Automatically designing CNN architectures using the genetic algorithm for image classification," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3840–3854, Sep. 2020.

[61] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Evolving deep convolutional neural networks for image classification," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 394–407, Apr. 2020.

[62] Y. Sun, B. Xue, M. Zhang, and G. G. Yen, "Completely automated CNN architecture design based on blocks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 4, pp. 1242–1254, Apr. 2020.

[63] Y. Sun, H. Wang, B. Xue, Y. Jin, G. G. Yen, and M. Zhang, "Surrogate-assisted evolutionary deep learning using an end-to-end random forest-based performance predictor," *IEEE Trans. Evol. Comput.*, vol. 24, no. 2, pp. 350–364, Apr. 2020.

[64] A. Martín, V. M. Vargas, P. A. Gutiérrez, D. Camacho, and C. Hervás-Martínez, "Optimising convolutional neural networks using a hybrid statistically-driven coral reef optimisation algorithm," *Appl. Soft Comput.*, vol. 90, May 2020, Art. no. 106144.

[65] A. Rawal and R. Miikkulainen, "From nodes to networks: Evolving recurrent neural networks," 2018, *arXiv:1803.04439*. [Online]. Available: http://arxiv.org/abs/1803.04439

[66] T. Kim and S. Cho, "Particle swarm optimization-based CNN-LSTM networks for forecasting energy consumption," in *Proc. Congr. Evol. Comput. (CEC)*, 2019, pp. 1510–1516.

[67] N. Xue, I. Triguero, G. P. Figueredo, and D. Landa-Silva, "Evolving deep CNN-LSTMs for inventory time series prediction," in *Proc. Congr. Evol. Comput. (CEC)*, 2019, pp. 1517–1524.

[68] K. O. Stanley, J. Clune, J. Lehman, and R. Miikkulainen, "Designing neural networks through neuroevolution," *Nature Mach. Intell.*, vol. 1, no. 1, pp. 24–35, Jan. 2019.

[69] P. Niu, S. Niu, N. Liu, and L. Chang, "The defect of the Grey wolf optimization algorithm and its verification method," *Knowl.-Based Syst.*, vol. 171, pp. 37–43, May 2019.

[70] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems," *Knowl.-Based Syst.*, vol. 96, pp. 120–133, Mar. 2016.

[71] Z. Gao and J. Zhao, "An improved Grey wolf optimization algorithm with variable weights," *Comput. Intell. Neurosci.*, vol. 2019, Jun. 2019, Art. no. 2981282.

[72] N. Mittal, U. Singh, and B. Sohi, "Modified Grey wolf optimizer for global engineering optimization," *Appl Comput. Intell. Soft Comput.*, vol. 2016, May 2016, Art. no. 7950348.

[73] W. Long, J. Jiao, X. Liang, and M. Tang, "An exploration-enhanced Grey wolf optimizer to solve high-dimensional numerical optimization," *Eng. Appl. Artif. Intell.*, vol. 68, pp. 63–80, Feb. 2018.

[74] Y. Gao, X. An, and J. Liu, "A particle swarm optimization algorithm with logarithm decreasing inertia weight and chaos mutation," in *Proc. Int. Conf. Comput. Intell. Secur. (CIS)*, 2008, pp. 61–65.

[75] R. A. Khanum, M. A. Jan, A. Aldegheishem, A. Mehmood, N. Alrajeh, and A. Khanan, "Two new improved variants of Grey wolf optimizer for unconstrained optimization," *IEEE Access*, vol. 8, pp. 30805–30825, 2020.

[76] X. S. Yang, "Random walks and optimization," in *Nature-Inspired Optimization Algorithms*. Oxford, U.K.: Elsevier: 2014, pp. 45–65.

[77] T. Kim and H. Y. Kim, "Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data," *PLoS ONE*, vol. 14, no. 2, Feb. 2019, Art. no. e0212320.

[78] J. Wang, W. Xu, X. Fu, G. Xu, and Y. Wu, "ASTRAL: Adversarial trained LSTM-CNN for named entity recognition," *Knowl.-Based Syst.*, vol. 197, Jun. 2020, Art. no. 105842.

[79] W. Li, L. Zhu, Y. Shi, K. Guo, and E. Cambria, "User reviews: Sentiment analysis using lexicon integrated two-channel CNN–LSTM family models," *Appl. Soft Comput.*, vol. 94, Sep. 2020, Art. no. 106435.

[80] J. Wang, L.-C. Yu, K. R. Lai, and X. Zhang, "Tree-structured regional CNN–LSTM model for dimensional sentiment analysis," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 28, pp. 581–591, 2020.

[81] J. Gehring, M. Auli, D. Grangier, and Y. N. Dauphin, "A convolutional encoder model for neural machine translation," 2016, *arXiv:1611.02344*. [Online]. Available: http://arxiv.org/abs/1611.02344

[82] X. Pan, G. Ying, G. Chen, H. Li, and W. Li, "A deep spatial and temporal aggregation framework for video-based facial expression recognition," *IEEE Access*, vol. 7, pp. 48807–48815, 2019.

[83] S. Ding, S. Qu, Y. Xi, and S. Wan, "Stimulus-driven and concept-driven analysis for image caption generation," *Neurocomputing*, vol. 398, pp. 520–530, Jul. 2020.

[84] L. Rodríguez, O. Castillo, J. Soria, P. Melin, F. Valdez, C. I. Gonzalez, G. E. Martinez, and J. Soto, "A fuzzy hierarchical operator in the Grey wolf optimizer algorithm," *Appl. Soft Comput.*, vol. 57, pp. 315–328, Aug. 2017.

[85] R. Cheng and Y. Jin, "A competitive swarm optimizer for large scale optimization," *IEEE Trans. Cybern.*, vol. 45, no. 2, pp. 191–204, Feb. 2015.

[86] C. Blake. (1998). *UCI Repository of Machine Learning Databases*. [Online]. Available: http://www.ics.uci.edu/?mlearn/MLRepository.html

[87] M. Q. Raza and A. Khosravi, "A review on artificial intelligence based load demand forecasting techniques for smart grid and buildings," *Renew. Sustain. Energy Rev.*, vol. 50, pp. 1352–1372, Oct. 2015.

[88] S. Zhang, B. Guo, A. Dong, J. He, Z. Xu, and S. X. Chen, "Cautionary tales on air-quality improvement in Beijing," *Proc. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 473, no. 2205, Sep. 2017, Art. no. 20170457.

[89] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proc. 21th Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn. (ESANN)*, 2013, pp. 437–442.

[90] X. Zhang, L. Yao, X. Wang, J. Monaghan, D. Mcalpine, and Y. Zhang, "A survey on deep learning based brain computer interface: Recent advances and new frontiers," 2019, *arXiv:1905.04149*. [Online]. Available: http://arxiv.org/abs/1905.04149

[91] Z. Wu, X. Wang, Y. G. Jiang, H. Ye, and X. Xue, "Modelling spatial–temporal clues in a hybrid deep learning framework for video classification," in *Proc. ACM Int. Conf. Multimed (ACM-MM)*, 2015, pp. 461–470.

[92] T.-H. Wen, M. Gasic, N. Mrksic, P.-H. Su, D. Vandyke, and S. Young, ''Semantically conditioned LSTM-based natural language generation for spoken dialogue systems,'' 2015, *arXiv:1508.01745*. [Online]. Available: http://arxiv.org/abs/1508.01745

[93] B. Fielding and L. Zhang, ''Evolving image classification architectures with enhanced particle swarm optimisation,'' *IEEE Access*, vol. 6, pp. 68560–68575, 2018.

[94] T. Y. Tan, L. Zhang, and C. P. Lim, ''Adaptive melanoma diagnosis using evolving clustering, ensemble and deep neural networks,'' *Knowl.-Based Syst.*, vol. 187, Jan. 2020, Art. no. 104807.

[95] T. Y. Tan, L. Zhang, C. P. Lim, B. Fielding, Y. Yu, and E. Anderson, ''Evolving ensemble models for image segmentation using enhanced particle swarm optimization,'' *IEEE Access*, vol. 7, pp. 34004–34019, 2019.

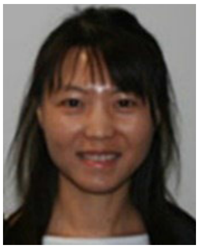
[96] P. Kinghorn, L. Zhang, and L. Shao, ''A region-based image caption generator with refined descriptions,'' *Neurocomputing*, vol. 272, pp. 416–424, Jan. 2018.

[97] M. A. Tawhid and A. M. Ibrahim, ''A hybridization of Grey wolf optimizer and differential evolution for solving nonlinear systems,'' *Evolving Syst.*, vol. 11, no. 1, pp. 65–87, Mar. 2020.

[98] N. Singh and S. B. Singh, ''A novel hybrid GWO-SCA approach for optimization problems,'' *Eng. Sci. Technol., Int. J.*, vol. 20, no. 6, pp. 1586–1601, Dec. 2017.

HAILUN XIE received the B.Sc. degree in built environment and energy application engineering and the M.Sc. degree in building service and energy conservation engineering from Shenyang Jianzhu University, China. He is currently pursuing the Ph.D. degree with the Department of Computer and Information Sciences, Northumbria University.

His research interests include machine learning, deep learning, and evolutionary computation.



LI ZHANG (Senior Member, IEEE) received the Ph.D. degree from the University of Birmingham, U.K.

She is currently an Associate Professor and a Reader of computer science with Northumbria University, U.K., and also serving as an Honorary Research Fellow at the University of Birmingham. She holds expertise in machine learning, deep learning, and evolutionary computation. She has served as an Associate Editor for *Decision Support Systems*.

CHEE PENG LIM received the Ph.D. degree from The University of Sheffield, U.K., in 1997. He is currently a Professor of complex systems with Deakin University, Australia. He has published over 450 technical articles in books, international journals, and conference proceedings. His research interests include computational intelligence, decision support, pattern recognition, medical prognosis and diagnosis, and fault detection and diagnosis.