**Jilin Daxue Xuebao (Gongxueban)/Journal of Jilin University (Engineering and Technology Edition)**
ISSN: 1671-5497
E-Publication: Online Open Access
Vol: 43 Issue: 08-2024
DOI: 10.5281/zenodo.13284361

# CONSTRUCTION OF ATTENTION BASED GRU MODEL WITH EFFECTIVE FEATURE SELECTOR FOR WEATHER FORECASTING

**YUKTI VARSHNEY**

Research Scholar, Department of Computing Science, Teerthanker Mahaveer University, Moradabad, India. Email: yuktivarshney16@gmail.com

**NUPA RAM CHAUHAN**

Associate Professor, Department of Computing Science, Teerthanker Mahaveer University, Moradabad, India. Email: nrcua80@gmail.com

### Abstract

Weather prediction is an appealing but demanding endeavor because of its substantial effects on human existence and the complex dynamics of atmospheric movement. Importance of weather forecasting is huge in daily life activities, business, agriculture etc. so; scholarly genre is taking great interest in this field. In weather prediction large amounts of data have been collected from multiple sources such as satellites, weather stations, radar and historical records, is complex task to process. Machine learning and deep learning approaches that rely on a huge amount of data with quickly and accurately become more popular. Numerous technique focus only temporal pattern of meteorological data, ignoring the correlations between multiple variables at various geographical locations. In this paper Chaotic Logistic Map Based Grey Wolf Optimization (CLMGWO) determine appropriate climate factor for each geographical location and Attention based Gated Recurrent Unit (AttGRU) provide a precise prediction of feature correlation with many parameter and station across temporal time stamp. Proposed method AttGRU_CLMGWO resolve the problem of feature selection with successfully capture concealed spatial interconnections and a wide range of enduring weather patterns. Finally AttGRU implemented with Root Mean Square Propagation (RMSProp) Optimizer to evaluation of mean square error. This error is used to recalibrate the weight and bais in order to get improved result. AttGRU_CLMGWO model is comprised with Graph Neural Network (GNN), Bayesian Multi-head Attention Encoder-Decoder Neural Network (BMAE-Net) and Convolutional Neural Networks (CNN). Proposed model AttGRU_CLMGWO has implemented in python using Jena Climate dataset and predict temperature and humidity by concurrently acquire data for crucial time stamp and weather forecasting. The results yielding from the AttGRU_CLMGWO is MSE - 1.3, an MAE - 0.41, and a MAPE - 0.2.

**Keywords:** Weather Forecasting, Feature Selection, Gated Recurrent Network, Time Series, Wind Direction.

## 1. INTRODUCTION

The weather prediction is a crucial application of scientific computing. It can predict future weather fluctuations, especially severe weather events like floods, droughts, and hurricanes, which is important for society (including daily activities, agriculture, energy production, transportation, industry, etc.). Over the last ten years, there has been significant progress in the scientific area of numerical weather prediction (NWP) due to the advancement of high-performance computing devices [1]. Traditional NWP approaches typically adhere to a simulation-based approach. This entails utilizing numerical simulations to solve partial differential equations (PDEs) representing the physical rules driving atmospheric conditions [2, 3, 4]. These NWP techniques often have low processing performance due to the complex nature of solving PDEs. For instance, calculating a single simulation for a 10-day forecast with a spatial resolution of 0.25◦×0.25◦ would take several hours on a supercomputer with hundreds of nodes [5]. This restricts the number of ensemble members that can be employed for probabilistic weather predictions and greatly reduces the timeliness of daily weather forecasts. Furthermore, traditional NWP algorithms heavily depend on parametric numerical models. However, despite their high complexity, these models are sometimes deemed insufficient [6, 7]. For instance, mistakes might arise due to the parameterization of unresolved processes. To tackle the aforementioned challenges, a potential approach involves using artificial intelligence, namely deep learning, to develop data-driven weather forecasting methods. Deep neural networks are used to capture the association between observed input data and anticipated output data. AI-based techniques may balance model complexity, prediction resolution, and accuracy on GPUs for fast performance [8], [9]. The spatial resolution of FourCastNet [10] has been increased to 0.25◦ x 0.25◦, equivalent to the ECMWF Integrated Forecast Systems (IFS). It generates a 100-member, 24-hour forecast in 7 seconds utilising four GPUs. This is orders of magnitude quicker than typical NWP approaches. Nevertheless, the FourCastNet's prediction accuracy is still unsatisfactory. The RMSE of the 5-day Z500 prediction using a single model and a 100-member ensemble is 484.5 and 462.5, respectively. These values are substantially lower than ECMWF's operational IFS of 333.7 [11]. It is hypothesised that many significant advancements are required before artificial intelligence (AI) technologies may surpass NWP. The majority of weather prediction systems were constructed based on the study or reanalysis of data beyond direct observations. The reanalysis datasets are often regarded as the most accurate estimates [12], [13] for most atmospheric variables, with the exception of some elements such as precipitation. This work uses ERA5, the 5th ECMWF reanalysis dataset [14]. Latitude, longitude, pressure levels (height), and time comprise the ERA5 dataset. We have the freedom to choose any number of weather parameters (such as geopotential, temperature, etc.), but we should not see them as contributing to a new dimension. The dataset, which has a size exceeding 2 petabytes (PB), is divided into two-dimensional (2D) slices based on latitude and longitude. This division is done to facilitate the process of downloading. However, by defining a time point (hourly for the previous 60 years), pressure level (or Earth's surface), and weather component, a matrix of global reanalysis data may be obtained. The overall ERA5 data is A. Superscripts denote meteorological variables and pressure levels, whereas subscripts provide spatiotemporal locations. Example: AT850 t shows global temperature data in matrix form at time t and height 850hPa. For geopotential data at point (x, y), time t, and 500hPa height, see AZ500 x,y,t. It is important to note that AZ500 x,y,t is a single numerical value. In order to mitigate the aforementioned load, researchers initiated a secondary investigation that explores AI techniques for weather forecasting. Deep learning enables the direct learning of complicated functions (represented by f(·)) from large amounts of training data, without requiring knowledge of the underlying physical procedures or formulas. Many deep neural networks describe f(·) as f(·; θ), where · is the input data and θ is the adjustable parameters. The Computer Vision (CV) analyses 2D/3D cubes of image data, making it the closest to weather forecasting.

Over the last ten years, the CV community has created numerous successful network architectures, such as those mentioned in references [16] and [17]. More recently, they have adapted powerful architectures called transformers from the field of natural language processing [18] and have developed variants [19] that can effectively handle image data. AI-based approaches were first used in weather forecasting to address the challenges of predicting future weather data in settings where traditional NWP methods, such as radar or satellite data-based precipitation forecasting, are inadequate [20]. The remarkable capacity of deep neural networks to convey information effectively has contributed to their success in data-driven environments. This success has motivated researchers to investigate the challenges faced by NWP methods, such as the significant computational burden associated with direct medium-range weather forecasting. In fact, this computational task has consumed a substantial portion of the computational resources of weather forecast centers over the past decade. The following are the contributions made by this work:

- First of all the globally optimize and efficiently anticipate the selection of parameters using the Grey wolf optimization algorithm, chaos has been employed to produce the chaotic grey wolf optimization algorithm.
- Chaotic maps are used in optimizing algorithms to enhance efficiency by effectively analyzing the search area, taking into account the nature of dependability. The findings of the efficient parameter indicate that Chaotic Logistic Map Gray Wolf Optimization (CLMGWO) outperforms traditional GWOs in terms of convergence when compared to other methods and applications.
- An attention-based multilayered GRU model has been developed with Root Mean Square Propagation (RMSProp) optimizer to evaluation of mean square error for improve the speed and stability of multi-step weather prediction. This model outperforms with other deep learning architectures and has been extensively compared to existing forecasting models, demonstrating its superiority.

The paper is structured as follows: Section 1 comprises the project's introduction. Section-2 provides a concise summary of the conducted literature survey. Section-3 elucidates the operational procedures of the system. Section-4 portrays the inferences and results obtained. The "Conclusion" section in section 5 summarizes our findings and discusses potential next directions.

## 2. RELATED WORKS

Deep Learning (DL) has been used in recent years to investigate time series difficulties [21], in which the relationship between characteristics is apparent but challenging to discern. Traditional machine learning techniques may not work as well for systems whose behavior is primarily impacted by temporal or geographical context, like weather systems. In contrast, DL methods, which can automatically extract spatio-temporal features, are more suitable for gaining a deeper understanding of such systems.

Improved prediction accuracy may be achieved by accurately analysing the association and appropriately representing the information. As a result, DL has been accepted as a sensible and adaptable technique for analysing time series characteristics. Consequently, several scientists have used DL techniques for the purpose of weather prediction, which is a common and complex issue involving multi-dimensional time series data. Data-driven solutions are anticipated to tackle some traditional challenges in weather forecasting. In [22] introduces the weather predicting model based on graph neural networks (GNNs) to analyse the data produced by these sensors. Graph learning-based models, or GNNs, perform well empirically in a variety of machine learning techniques.

A new neural network architecture, BMAE-Net, is described in [23]. A Bayesian inference-optimized multi-head attention encoder-decoder framework is used. The main objective of BMAE-Net is to properly anticipate weather time series changes. Bayesian inference is added to the gated recurrent unit to generate the Bayesian-gated module. Next, each Bayesian layer's network architecture includes a multi-head attention mechanism to increase time duration prediction. Following that, Bayesian hyperparameter optimisation is used to create an encoder-decoder system. This framework deduces massive time-series data's underlying links for reliable forecasts. A deep learning method, multivariate data decomposition approach, grid search algorithm, and attention mechanism are used to create a hybrid wind speed prediction model based on weather research and forecasting (WRF) simulation [24].

In [25] proposes an optimised stacked Bi-directional Long Short-Term Memory (BiLSTM)/ LSTM model to forecast univariate and multivariate hourly time series data using stacked LSTM layers, drop out architecture, and LSTM-based model. By tweaking six pertinent hyperparameters, Bayesian optimisation improves the model's performance. In order to anticipate many fundamental atmospheric variables on a worldwide grid, [26] offer a notably enhanced data-driven global weather forecasting system that makes use of a deep CNN. With just a few input atmospheric condition variables, it may be trained to predict intricate patterns of surface temperature. The goal of [27] is to convert a weather forecasting system using deterministic neural networks into an ensemble model. We evaluate four approaches to build the ensemble: using random dropout in the network, retraining the neural network, creating early perturbations using singular vector decomposition, and random beginning perturbations. In [28], a method for forecasting future temperature using a neural network based on historical temperature data is proposed. To be more precise, authors developed a CRNNmodel, which consists of a CNN and a RNN.

While numerical models are now indispensable, they are costly to execute and include several physical phenomena that cannot be well represented by equations. The issue of error propagation during model solution is a significant factor that leads to inaccurate predictions. Hence, researchers are now prioritizing the advancement of a data-centric weather forecasting system that guarantees optimal efficacy and affordability, while delivering superior precision and reliability in weather forecasts.

### 2.1 Chaotic Grey Wolf Optimization (CGWO)

Gray Wolf Optimization (GWO) is a meta-heuristic approach that draws inspiration from grey wolf hunting behavior and social structure. As apex predators, grey wolves often live in packs of five to twelve members, with a rigid hierarchy of dominance. Grey wolves hunt mostly by monitoring their prey, pursuing it, and then approaching it to annoy it until it stop. Drawing on the aforementioned information, an altered GWO algorithm that employs four operators to protect against the wolves becoming stuck at the ideal local point: Chaotic map, Opposition-based learning, Differential Evolution operators, and Disruption operator. A chaotic opposition based approach to choose the most appropriate starting population. Additionally, because DE operators function as a local search mechanism, we use them to enhance the wolves' capacity to exploit the region in their neighborhood. The population's diversity is essential to improving the search process; the disruption operator keeps the population diverse while enhancing the wolves' capacity for exploitation. As a result, the suggested approach accounts for every element that has an impact on the GWO's performance. These studies try to improve the capacity for exploration and exploitation. There are, however, modified GWO approaches that focus on diversity or beginning population selection; nonetheless, no method can take into account all of these elements at once [29].

### 2.2 Attention-Based Gated Recurrent Unit (AttGRU):

One of the most useful model in deep learning is Attention Based mechanism, which tell to an advancement of encoder–decoder technique, and were created to move forward the execution of long input arrangements. In Attention mechanism, the decoder can specifically get to the encoded data and employments a modern concept for the setting vector, which is presently calculated at each time step of the decoder, from the past covered up state and all the covered up states of the encoder. Trainable weights will be allotted out to these states and deliver diverse degrees of significance to all the components within the input arrangement.

An improved Attention-Based mechanism that weighs different value with the GRU is proposed. Finally, the improved Attention-Based mechanism is combined with GRU, and CRS is used to generate the optimal parameter combination at the attention layer. As a result, Attention-Based Gated Recurrent Unit (AttGRU) model captures long-term impact and higher the degree of attention that GRU pays to the function of sub-windows in various factors. The proposed Attention-Based Gated Recurrent Unit (AttGRU) model in this study combines the attention mechanism with GRU. On the basis of the ability of GRU to handle time series prediction problems [30].

## 3. MATERIALS AND PROPOSED METHODS

### 3.1 System model

The initial step is loading the Jena dataset, which contains a collection of meteorological time series data. The datasets are preprocessed using conventional methods. In this paper, the wind direction is transformed from degrees to axis and the time stamp is converted from day to year. The preprocessed data is subjected to feature selection using the Chaotic Logistic Map Based Grey Wolf Optimisation algorithm, which effectively optimises and predicts the parameters on a global scale on time series pattern.  As shown in figure-1, the chosen features are fed into the Attention Based GRU Model which decoder can specifically get to the encoded data for calculating each time stamp, and trainable weight is allotted to these state and delivers the diverse degree of arrangement. After that AttGRU work RMSProp Optimizer in order to predict the evaluation matrix for the error correction. All the error can be reduced with the help of RMSProp to predict the weather condition.

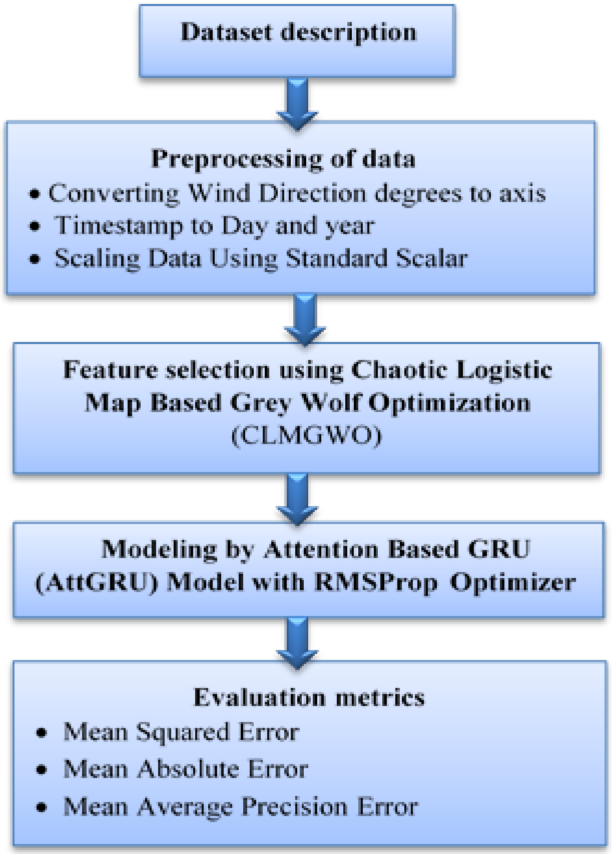
Dataset description
Preprocessing of data
• Converting Wind Direction degrees to axis
• Timestamp to Day and year
• Scaling Data Using Standard Scalar
Feature selection using Chaotic Logistic Map Based Grey Wolf Optimization (CLMGWO)
Modeling by Attention Based GRU (AttGRU) Model with RMSProp Optimizer
Evaluation metrics
• Mean Squared Error
• Mean Absolute Error
• Mean Average Precision Error

**Figure 1: Block Diagram of AttGRU_CLMGWO**

(Attention Based Gated Recurrent Unit with Chaotic Logistic Map Based Grey Wolf Optimization)

### 3.2 Dataset description

Jena Climate is a collection of meteorological time series data acquired by the weather station located at the Max Planck Institute for BioGeochemistry in Jena, Germany. The collection comprises 14 distinct observations collected at 10-minute intervals across many years, including air temperature, atmospheric pressure, humidity, and wind direction. The dataset includes data spanning from January 1st, 2009 to December 31st, 2016, consisting of 420,451 data points [31]. This research examines the periodic regularity of temperature fluctuations throughout months and hours. It aims to map and establish rules for temperature changes within a 24-hour period for each month. There are variations in temperature across various months, and three components - month, cos (h), and sin (h) - are included in the dataset to represent these variations. The trigonometric function of hours is used to ensure that the same pattern repeats itself every 24 hours. This research utilises data collected between 2014 and 2016, specifically picking the variables 'T (degC)', 'p (mbar)', 'rh (%)', and 'H2OC (mmol/mol)' from a pool of 14 original quantities. Additionally, three time-related parameters are included. In all, there are 157,824 data points containing seven variables, which will be used for prediction purposes.  Data must be standardised due to distinctive element distribution and positive and negative values. Data was divided into three 6:2:2 subsets: training, validation, and testing.

## Preprocessing of data

When the test wind turbine is stationary, the wind speed at this test site fluctuates between the wake and reference wind measurement locations. Furthermore, the geographical setting has a significant influence on the wake of wind turbines. The non-dimensional wind velocity ratio $U_{NR}$ removes wind velocity and wake distribution changes from wake measurement. The test wind turbine's non-dimensional wind velocity $U_{no}$ is measured while operating and $U_{np}$ while motionless. The equation gives non-dimensional wind velocity ratio $U_{nr}$.

$$U_{nr} = \frac{U_{no}}{U_{np}} \qquad (1)$$

The values of $U_{no}$ and $U_{np}$ are averaged over all retrieved data.   Furthermore, the value of $U_{no}$ is determined by dividing the observed wake wind velocity $U_{wake\_o}$, obtained from sonic anemometers positioned on the wake measuring mast, by the reference wind velocity $U_{wake\_p}$ during the operation of the test wind turbine. The value of $U_{np}$ is determined by dividing the wake wind velocity $U_{wake\_p}$ by the reference wind velocity $U_{ref\_p}$ while the test wind turbine is not moving. The values of $U_{no}$ and $U_{np}$ are explicitly specified as

$$U_{no} = \frac{U_{wake\_o}}{U_{ref\_p}} \, U_{np} = \frac{U_{wake\_p}}{U_{ref\_p}} \qquad (2)$$

Weather impacts building energy usage and solar energy generation. Temperature changes affect heating and cooling needs, thus buildings with similar temperature patterns should have similar demands. Sunlight is the main energy source for PV systems and increases with temperature. Because of their importance, temperature and global horizontal irradiance (GHI) were taken into account while matching timestamps to reconstruct building load. We match each time stamp t with a collection of comparable timestamps θ with the aid of this instruction. Put differently, every period t is associated with a collection of similar timestamps$\theta_t = \{\theta_t\}$. All premises have identical settings. Since each premise k has a unique PV installation date, the time stamps are separated into two sets: pre- and post-installation. PV installation time is $t_{k,I}$. β represents time stamps before installation $t < t_{k,I}$ while α represents time stamps after installation $t > t_{k,I}$. In order to account for discrepancies between the reported and real dates for PV system activation, we also avoid from utilizing time stamps that are set to expire on the day of installation. As a buffer in this work, b = a = 20 days, hence for each premise k, the β and α are defined as

$$\beta_k = \{t : t \epsilon t < (t_{k,I} - b)\} \qquad (3)$$

We determine the collection of similar timestamps after installation for each $\beta_{k,i} \epsilon \beta_k$

$$\varphi_{k,i}^{(\beta)} = \{\varphi_t : t \epsilon \beta_{k,i}\} \qquad (4)$$

Where parenthetical superscripts represent all premise k time stamps. These generate two 2D dictionaries with equivalent timestamps for each time stamp in k and k.

First and foremost, characteristics that are superfluous or that could include duplicate data must be eliminated. To achieve this objective, we used the values of the covariance matrix. Several humidity and temperature metrics were removed from the dataset as a consequence of this investigation because there were too many zeros, the column lights were also removed. The filters were employed after a comprehensive study, including factors such as correlation, zeros, null values, and other relevant discoveries. Furthermore, it is essential to scale the characteristics of the dataset prior to training the model. This is due to the limited efficacy of the current models within a narrower numerical spectrum. Optimizers can readily identify the learning rate, which creates a favourable setting for testing. Equation (5) illustrates the process of scaling the dataset, which is a necessary pre-processing step before to training. The term $X'$ denotes the dataset that has been scaled. The data is scaled down inside a discrete set, ranging from -1 to 1, using minmax scalar, and then retrieved.

The scalar value x represents a value taken from a feature vector $X$. $x_{min}$ and $x_{max}$ correspond to the lowest and maximum scalar values, respectively, obtained from the feature vector $X$. By using Equation (5)

$$x' = \frac{(x - x_{min})(max - \min)}{x_{max} - x_{min}} + \min \quad x \epsilon X \qquad (5)$$

Furthermore, the weight factors for each characteristic are computed. This is achieved by using the scaled data to evaluate several models in order to determine the optimal model configurations and training parameters.

### 3.3 Proposed methodology for Chaotic Logical Map based GWO

- Proposed methodology implements Chaotic Logical Map based algorithm with a modified Grey Wolf Optimization (GWO) algorithm called CLM Grey Wolf Optimization (CLMGWO). It is used for feature selection.
- The selection of a subset of features which used to maximize classification accuracy. It minimizes the selected featured numbers.

- Grey wolf's population called "agents" (candidate solutions) is initialized randomly. A subset of features is represented by each agent.
- By using classification algorithm to measure their fitness and accuracy, agents are set to train and evaluate.
- Agents are guided by top hierarchy then sub hierarchy called alpha, beta, delta wolves for the movement of the pack (data).
- To get optimal features subsets, iterations of solutions are updated.
- For probabilistically toggle feature values in each agent's solution a transfer function is used.

### 3.3.1 Feature selection using CLMGWO

Feature selection involves the selection of a concise subset of characteristics that are both essential and adequate to accurately define the target notion. The appropriate feature set is essential for every learning algorithm since it's its only source of knowledge. The main goal of feature selection is to avoid selecting too many traits. If a limited number of features are chosen, it is quite likely that the information included in this collection of features is minimal. GWO is an optimization heuristic that utilizes the selection criteria of grey wolves. It was created by Mirjalili et al. in 2014. This algorithm is a meta-heuristic approach that is enhanced by the observed framework of hunting behaviors and social organization of grey wolves. Not every GWO search iteration can be global. Thus, finding the global best answer is sometimes necessary. There was consistent search functionality. The GWO strategy for prey seeking employs a hierarchical approach, including the encircling, hunting, attacking, and search for prey employing optimization techniques. The hierarchical technique divides wolves into four categories. The first three dominant wolves, α, β, and δ, guide and supervise the other wolves' hunting efforts. Grey wolves use a hunting strategy known as encircling, when they surround their victim and communicate the prey's location to one another. After the prey's location is identified, the hunting process is carried out by other wolves under the leadership of the leader wolves. The arrangement of the wolves around the prey is revised according on the guidance of leader wolves in order to determine the prey's location. The method of assaulting prey involves the process of exploitation. The search for prey involves an exploratory phase and terminates by abruptly deviating from the best option. Grey wolves use the surround prey technique to isolate their victim according to set guidelines while hunting. A flow diagram is illustrated for feature selection in GWO. This flowchart provides a visual representation of the Feature Selection using GWO algorithm, helping to understand the sequence of steps and decision-making in the process.

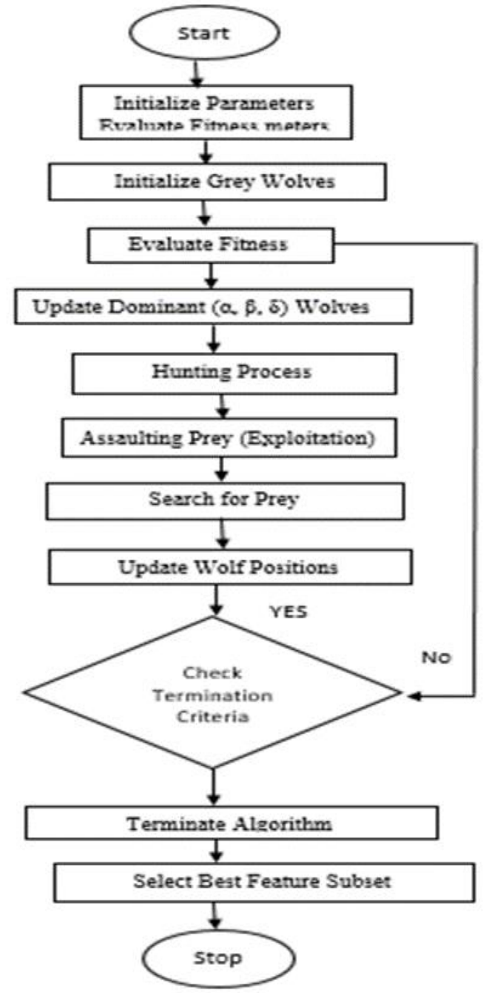

**Figure 2: Flow diagram for Feature selection using CLMGWO**

Algorithm for the proposed Grey Wolf Optimization (GWO) approach for feature selection

**Algorithm 1: Proposed CLMGWO training model**

```
Input: The weather data, hyper parameter space W, epochs
Output: the optimal hyper parameter, the prediction of temperature
1: Initialize weather parameters
2: Number of grey wolves (n)
3: Maximum iterations (p)
4: Training data
5: Objective function (e.g. classification accuracy)
6: Initialize population of n grey wolves (agents)
```

```
7: Each wolf is a d-dimensional vector representing a subset of d features
8: Identify best parameter 3 wolves as alphas (α), betas (β), and deltas (δ)
9: Leader wolves based on objective function
10: for i =1 to p do
11: Update positions of all followers using encircling, hunting, and attacking movement equations based on α, β, δ positions
12: Update grey wolf positions stochastically to explore search space
13: Evaluate new wolf positions using objective function and update α, β, δ leader wolves
14: Repeat step 10 for maximum iterations
15: Return best alpha wolf position
16: Represents selected feature subset
17: Obtained the best parameter and assess the performance of selected features
```

So in essence, GWO mimics the social hierarchy and hunting behavior of grey wolves to perform optimization search for finding the fittest feature subsets through iterative evaluation and evolution guided by leader wolves.

$$P' = \left|R'.Y'_p(t) - Y'(t)\right| \qquad (6)$$

$$Y'(t+1) = Y_p(t) - V'.P' \qquad (7)$$

The equations (6) and (7) correspond to the iteration number $t$. The labels $Y'_p$ and $Y'$ stand for the level of prey and grey wolf, respectively. $V' = 2a' r'_1 - a', R = 2r'_2$depend on the number of iterations and the random vectors of [0,1]. The features of the parameter $a'$ are successively lowered from 2 to 0. Both $r'_1$ and $r'_2$ are random vectors.  The hunting technique for capturing prey is carried out in accordance with the previous regulations.

$$P'_i = |R'_i.Y'_i(t) - Y_i(t)| \qquad (8)$$

Where, $i$ represent $\alpha$, $\beta$ and $\delta$

$$Y'(t+1) = \sum_{i=\{\alpha,\beta,\delta\}} Y'_i(t) - V'_i \, . P_i \qquad (9)$$

In equations (8) and (9), the variables $X_i$ represent the location of leader wolves $R'_i$, whereas $V'_i$ represents a random vector. The determination of assault and prey detection is shown by the vector $V, R'$.  An analysis is conducted when $A$ is larger than 1 or $A$ is less than -1. Otherwise, C is greater than 1. Conversely, the theft occurs when the absolute value of |V|<1 and |R|<1.

This study introduces a novel and effective strategy to enhance the Software reliability growth model (SRGM) metrics in order to enhance the searching behaviour of the GWO. A refined grey wolf optimization method is proposed, using an adaptable chaotic search approach to increase the search process and minimize the possibility of inaccurate predictions via the CGWO algorithm. The Different forms of chaotic maps provide chaotic variables for chaotic algorithms. The chaotic search strategy is introduced and characterized while considering the chaotic map.

$$Cx_i^{n+1} = \mu Cx_i^n(1 - Cx_i^n) \qquad (10)$$

In equation (10), the symbol $Cx_i^n$ represents the chaotic variable, while the symbol n represents the number of iterations. Researchers, mathematicians, and medical scientists have extensively used these chaotic maps in the area of optimisation. Efficiently navigating the search is clearly beneficial.

Initial chaotic maps are [0, 1]. Statistical study of the literature gives these maps a starting value of 0.7. Every chaotic map has a unique attitude. Chaos maps help determine data. Every user-defined grey wolf in the target zone is checked for fitness and categorised by condition using standard benchmarking metrics. The issue has goal and constraint violation functions. Formulate minimal problems. It might be phrased as

$$minQ(y), y = (y_1, y_2, y_3, \dots y_n)\epsilon P^n \qquad (11)$$

In equation (11), the variable $n$ represents the estimated number of configurations of a feasible solution. The symbol $Y\epsilon Q\epsilon S, Q$ indicates that $Y$ belongs to the potential area $Q$ inside the search space S. Furthermore, $Q$ is defined as an n-dimensional rectangle. The domain of $P$ is characterised by a lower bound (l) and an upper bound (u), as specified in Equation (12). The $Q$ space, stated in Equation (13), represents the range of limitations $(g > 0)$ in P.

$$l(j) \leq y(j) \leq u(j), \quad 1 \leq j \leq n \qquad (12)$$

$$r_k(y) \leq 0, for\ k = 1,2, \dots v \qquad (13)$$

$$s_k(y) = 0, for\ k = v + 1, \dots . g \qquad (14)$$

If a solution in $Q$ space satisfies either the constraint $r_u$ or $s_u$, then $r_u$ is considered an active limitation at $y$ in equations (13) and (14), whereas $r_k(\mathrm{y})$ and $s_k(\mathrm{y})$ are regarded as inequality and equality restrictions, correspondingly.

The suggested approach, known as the chaotic GWO algorithm, is used to effectively solve optimization issues. The program begins by initializing a population of wolves. The chaotic map value is initialized as $x_0$ and then updated continually.

The parameters $a', A', C'$ are sequentially engaged in carrying out the exploration and exploitation procedure. The variable $t$ represents the number of iterations. The fitness of each search wolf is assessed using the variables $x_\alpha, x_\beta$ and $x_\delta$.

The first wolf appears as α, the second as β, and the third as δ. Grey wolves are categorised by iterative fitness. Chaotic map equations increase chaotic number. For every search wolf, both the location and the parameter value are changed, as stated in (10). Next, the poorest fit wolf is replaced with the best fit wolf. The CLMGWO method demonstrates the ideal solution by observing the fitness of an alpha wolf towards the end of the iteration. This strategy yields superior outcomes and efficiently saves computational resources.

### 3.3.2 Proposed Methodology for building an attention-based GRU model:

- To compute attention, vector an Attention Layer custom layer is defined over the GRU hidden state sequence.
- This layer is consisted of two main parts: Computing an attention score between individually hidden state then a context vector.
- To compute a weighted average context vector, the attention scores is used.
- The entire context from the sequence is condensed by the output attention vector and fed through a dense layer for obtaining desired results.
- return_sequences=True is used to process the input sequence.
- To compute the attention vector for Attention Layer, GRU layer is wrapped.
- A final Dense layer with 1 unit makes a prediction based on the attention vector and model is accumulated with the RMSProp optimizer and mean square error.

An artificial neural network (ANN) is a widely used AI method that emulates the functioning of human neurons to handle vast quantities of input simultaneously and learn effectively. ANNs are deterministic models that ignore time and focus on input and output variables. Using the input and weight vectors, the output is internally determined. Weight vector and decision boundary are perpendicular. An activation function affects the perceptron's input response, giving the ANN various decision bounds. However, a recurrent neural network (RNN) may dynamically map inputs to outputs, taking all time steps into account. RNN are particularly suitable for analysing time-series data due to their ability to sequentially analyse the input, while retaining an internal state that carries information from one-time step to the next. The most popular and effective RNN is the LSTM. To avoid RNN long-term reliance, the LSTM preserves differential input values during backpropagation. The RNN version GRU simplifies LSTM structure by lowering hidden state update computation. Figure-2 shows that it solves long-term reliance and preserves LSTM performance.

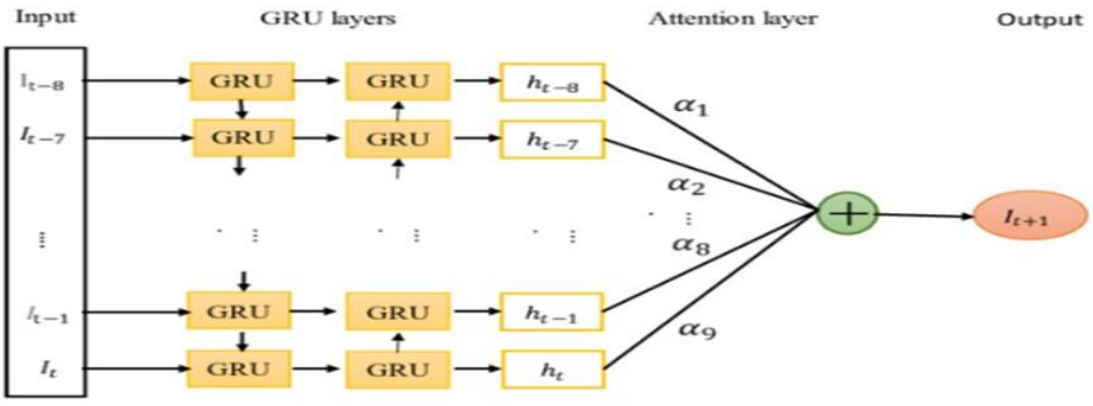

**Figure 3: Attention Based GRU Model**

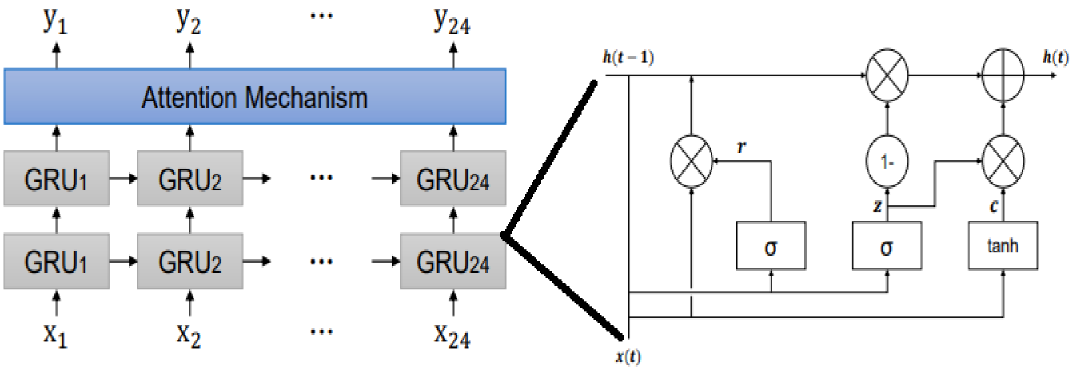

**Figure 4: Architecture of Attention Based GRU Model**

There are input and forget gates installed in the GRU cell. The input and forget gates are controlled by the gate controller, represented by the letter z. The input gate is open when z = 1, while the forget gate is closed otherwise. The input gate is closed and the forget gate is open when the value of z equals zero. Every iteration stores the previous (t-1) memory and resets the current time step's input. The following equations control the GRU cell: (15) – (18).

$$r_t = \sigma(W_r h_{t-1} + U_r x_t) \qquad (15)$$

$$z_t = \sigma(W_z h_{t-1} + U_z x_t) \qquad (16)$$

$$c_t = tan\,h(W_c(h_{t-1} \times r) + U_c x_t) \qquad (17)$$

$$h_c(z \times c) + ((1 - z) \times h_{t-1}) \qquad (18)$$

In order to create our GRU model, we considered hyperparameters and used a GRU network to anticipate the weather at 24 distinct time periods (spanning from one hour to one day in the future). Two hidden layers made up the configuration of the GRU model. The GRU model has two hidden layers, each with 13 nodes. The number of hidden layers is equal to the product of the output layer's size and 2/3 of the input layer's nodes. Equation (19) gives the scaled exponential linear unit (SELU), which was the activation function used in our investigation. The stochastic variable denoted by α in this equation is chosen at random during training from a uniform distribution. In the course of testing, however, α is fixed at 1.67326, which is the distribution's expected value.  Furthermore, λ is an extra parameter that's utilised to calculate the slope; by default, its value is 1.0507. The reason SELU works so well for training deep learning models is because of its remarkable self-normalization abilities and its ability to use equation (20) with δ = 1 to get around the problem of vanishing gradients. The learning rate and learning epoch were configured as 0.001 and 500, correspondingly.

$$f(\alpha, \lambda, \delta) = \begin{cases} \lambda(\alpha e^{x} - \alpha) for\ x < 0 \\ \lambda \mathrm{x\ for\ x} \geq 0 \end{cases} \qquad (19)$$

$$L_\delta\big(y, f(x)\big) = \begin{cases} 1/2(f(x))^2 for\ |y - f(x) for\ x < 0| \\ \delta|y - f(x)| - \frac{1}{2\delta^2} otherwise \end{cases} \qquad (20)$$

When the GRU network processes a longer input sequence, the accuracy of the output sequence prediction decreases. Even though input variables may have different associations with the forecasting goal, the network handles them all equally. Attention mechanisms may concentrate on key input variables. Encoders create attention vectors from input, while decoders create hidden states from encoder output. The encoder assigns an attention score to each concealed state by utilising the decoder's hidden state from the previous viewpoint. Applying a soft-max function to the attention score generates an attention vector. Thus, the encoder prioritises related input variables when the decoder anticipates output.

## 4. RESULT AND PERFORMANCE ANALYSIS

The experimental data is analyzed using Python software, using the parameters of MSE, MAE and MAPE. The parameters are compared with four advanced methods: Graph Neural Networks (GNNs) [22], Bayesian inference strategy (BMAE-Net) [23], convolutional neural network (CNN) [26], and the proposed AttGRU_CLMGWO.

**Mean Square Error (MSE)**: The model's prediction using the MSE technique quantifies the difference between the actual observation and the estimated observation. The application of data enhances the model's prediction power to some extent without excluding any necessary variables. The Mean Squared Error (MSE) is expressed as:

$$MSE = \sum_{k=1}^{n}(q_k - q'_k)^2 \qquad (21)$$

In equation (21), $q_k$ is the total count of identified faults at the specific time $t_k$, using real data. $q'_k$ is the estimated total count of identified discrepancies at time $t_k$, using the number of observations in the dataset of software failures.

**MAE and MAPE:** The developed model's efficiency is determined using MAE and MAPE after the predicted values have been obtained. As model efficiency increases, error parameters should decrease. The expressions for the parameters are provided below.

$$MAE = \frac{1}{N}\sum_{k=1}^{N}|e_k| \qquad 22$$

$$MAPE = \frac{1}{N}\sum_{k=1}^{N}\frac{|y_k - y'_k|}{y_k} * 100\% \qquad 23$$

Where the error factor, denoted as $e_k$, represents the difference among the actual value $y_k$ and the anticipated value $y'_k$.

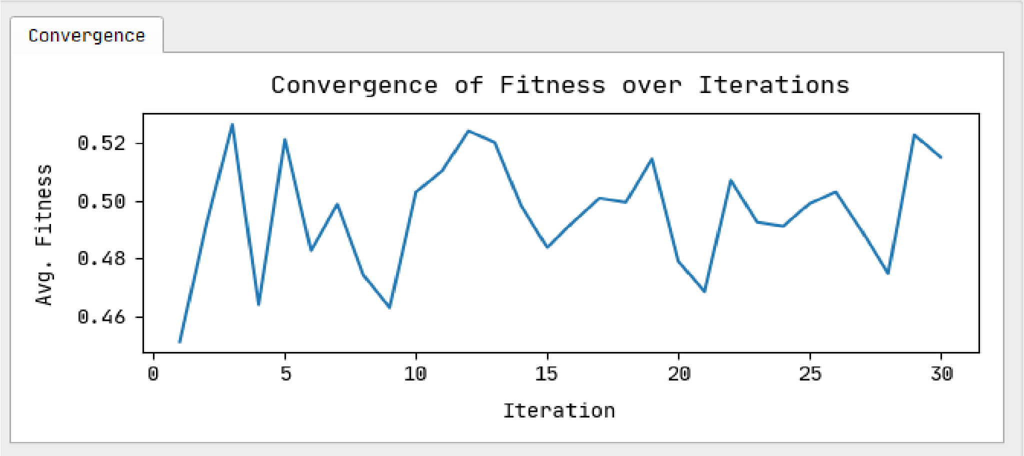

**Figure 5: Convergence of fitness over Iterations**

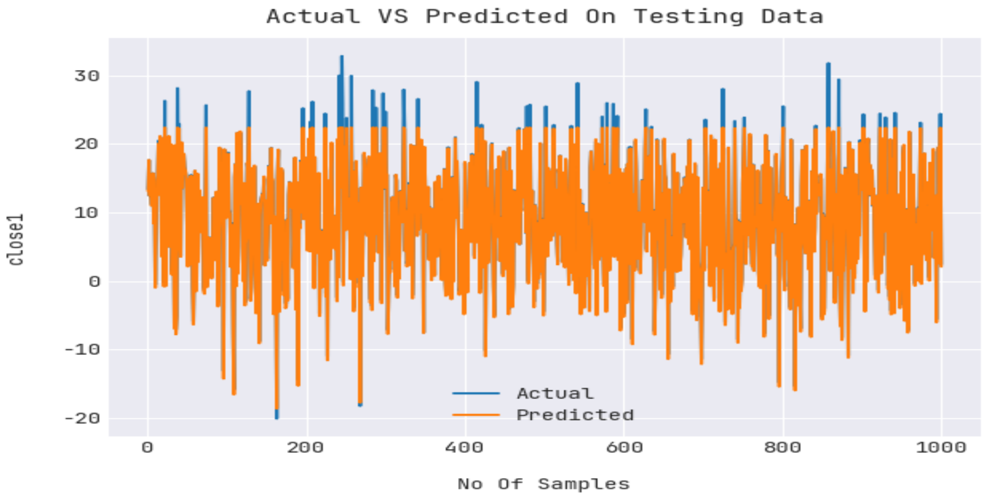

**Figure 6: Analysis of actual and predicted data during testing**

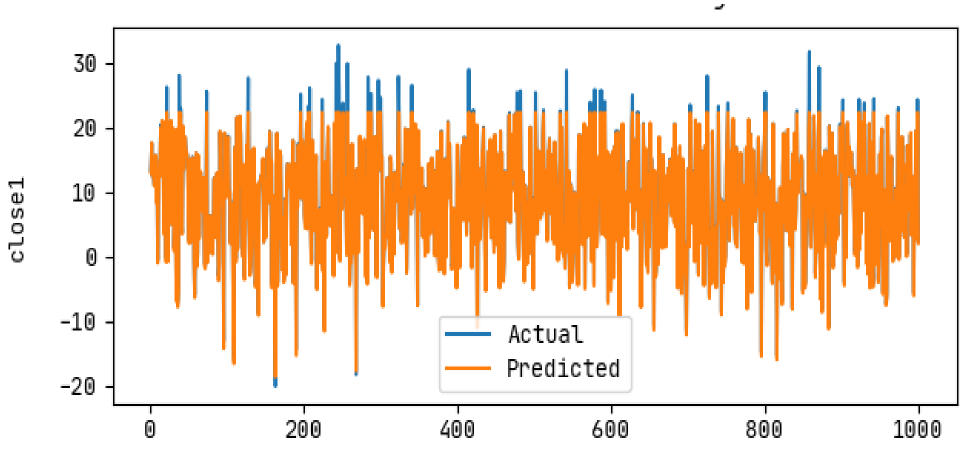

**Figure 7: Analysis of actual and predicted data during training**

Figure 6 illustrates the real and forecasted data throughout the testing assessment. The horizontal axis depicts the quantity of samples, and the vertical axis illustrates the close1 value. The greatest real value is obtained when the number of samples is either 200 or 1000 during the analysis. Figure 7 illustrates the real and forecasted data during the training assessment. The horizontal axis depicts the quantity of samples, and the vertical axis illustrates the close1 value. The greatest real value is achieved when the number of samples is either 200 or 800 during analysis.

**Table 1: Comparison of train and testing values for various metrics**

| Methods | Train | Test |
|---|---|---|
| MSE | 1.3601 | 1.2228 |
| MAE | 0.4194 | 0.4067 |
| MAPE | 0.265 | 0.2129 |

**Table 2: Comparative analysis between existing and proposed methods**

| Methods | GNNs [22] | BMAE-Net [23] | CNN [26] | AttGRU CLMGWO [proposed] |
|---|---|---|---|---|
| MSE | 3.5 | 2.6 | 4.3 | 1.3 |
| MAE | 4.2 | 3.7 | 4.2 | 0.41 |
| MAPE | 2.4 | 3.9 | 3.7 | 0.2 |

## 5. FUTURE SCOPE

The scope of this study is to further utilization of AttGRU_CLMGWO model. This model illustrated the iterative training methods and comparing the desired results continuously. This model is adjusting weights and biases too by aiming minimum error. The entire system supports to discern complex input into desired output. Through this study we have implemented humidity, temperature for hours and day. Although there are several potential directions of this study including by using predictive framework air density, barometric pressure can be measured effectively. Additionally, we can extend our study by extending various geographical regions or several climatic factors.

## 6. CONCLUSION

This study use AttGRU_CLMGWO for weather forecasting. The procedure employs an iterative training method, where by it consistently compares the observed output with the desired output and computes the error. This error is used to recalibrate the weights and bias in order to get an improved result. Therefore, this strategy aims to reduce the error. The system takes complicated factors as input and utilizes them to develop intelligent patterns during training. It then employs these patterns to make predictions. The input parameters considered for the predictions are temperature and humidity measurements from one hour and one day before, in addition to the seasonal factor. Moreover, this task may be expanded by using other factors like air density, precipitation, and barometric pressure to improve the precision of the forecasts and provide more complete weather predictions.

## References

1) P. Bauer, A. Thorpe, and G. Brunet, “The quiet revolution of numerical weather prediction,” Nature, vol. 525, no. 7567, pp. 47– 55, 2015.

2) W. C. Skamarock, J. B. Klemp, J. Dudhia, D. O. Gill, D. M. Barker, W. Wang, and J. G. Powers, “A description of the advanced research wrf version 2,” National Center For Atmospheric Research Boulder Co Mesoscale and Microscale . . . , Tech. Rep., 2005.

3) F. Molteni, R. Buizza, T. N. Palmer, and T. Petroliagis, “The ecmwf ensemble prediction system: Methodology and validation,” Quarterly journal of the royal meteorological society, vol. 122, no. 529, pp. 73–119, 1996.

4) H. Ritchie, C. Temperton, A. Simmons, M. Hortal, T. Davies, D. Dent, and M. Hamrud, “Implementation of the semi-lagrangian method in a high-resolution version of the ecmwf forecast model,” Monthly Weather Review, vol. 123, no. 2, pp. 489–514, 1995.

5) P. Bauer, T. Quintino, N. Wedi, A. Bonanni, M. Chrust, W. Deconinck, M. Diamantakis, P. Duben, S. English, J. Flemming ¨ et al., The ecmwf scalability programme: Progress and plans. European Centre for Medium Range Weather Forecasts, 2020.

6) T. Palmer, G. Shutts, R. Hagedorn, F. Doblas-Reyes, T. Jung, and M. Leutbecher, “Representing model uncertainty in weather and climate prediction,” Annual Review of Earth and Planetary Sciences, vol. 33, no. 1, pp. 163–193, 2005.

7) M. R. Allen, J. Kettleborough, and D. Stainforth, “Model error in weather and climate forecasting,” in ECMWF Predictability of Weather and Climate Seminar. European Centre for Medium Range Weather Forecasts, Reading, UK, 2002, pp. 279–304.

8) M. G. Schultz, C. Betancourt, B. Gong, F. Kleinert, M. Langguth, L. H. Leufen, A. Mozaffari, and S. Stadtler, “Can deep learning beat numerical weather prediction?” Philosophical Transactions of the Royal Society A, vol. 379, no. 2194, p. 20200097, 2021.

9) S. Scher and G. Messori, “Weather and climate forecasting with neural networks: using general circulation models (gcms) with different complexity as a study ground,” Geoscientific Model Development, vol. 12, no. 7, pp. 2797–2809, 2019.

10) J. Pathak, S. Subramanian, P. Harrington, S. Raja, A. Chattopadhyay, M. Mardani, T. Kurth, D. Hall, Z. Li, K. Azizzadenesheli et al., “Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators,” arXiv preprint arXiv:2202.11214, 2022.

11) P. Bougeault, Z. Toth, C. Bishop, B. Brown, D. Burridge, D. H. Chen, B. Ebert, M. Fuentes, T. M. Hamill, K. Mylne et al., “The thorpex interactive grand global ensemble,” Bulletin of the American Meteorological Society, vol. 91, no. 8, pp. 1059–1072, 2010.

12) A. K. Betts, D. Z. Chan, and R. L. Desjardins, “Near-surface biases in era5 over the canadian prairies,” Frontiers in Environmental Science, vol. 7, p. 129, 2019.

13) Q. Jiang, W. Li, Z. Fan, X. He, W. Sun, S. Chen, J. Wen, J. Gao, and J. Wang, “Evaluation of the era5 reanalysis precipitation dataset over chinese mainland,” Journal of hydrology, vol. 595, p. 125660, 2021.

14) H. Hersbach, B. Bell, P. Berrisford, S. Hirahara, A. Horanyi, ´ J. Munoz-Sabater, J. Nicolas, C. Peubey, R. Radu, D. Schepers ˜ et al., “The era5 global reanalysis,” Quarterly Journal of the Royal Meteorological Society, vol. 146, no. 730, pp. 1999–2049, 2020.

15) Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.

16) A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems, 2012.

17) K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Computer Vision and Pattern Recognition, 2016.

18) A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” Advances in neural information processing systems, vol. 30, 2017.

19) A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” arXiv preprint arXiv:2010.11929, 2020

20) V. Lebedev, V. Ivashkin, I. Rudenko, A. Ganshin, A. Molchanov, S. Ovcharenko, R. Grokhovetskiy, I. Bushmarinov, and D. Solomentsev, “Precipitation nowcasting with satellite imagery,” in Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, 2019, pp. 2680–2688.

21) J.N. Liu, Y. Hu, Application of feature-weighted support vector regression using grey correlation degree to stock price forecasting, Neural Comput. Appl. 22 (1) (2013) 143–152

22) Singh, G., & Durbha, S. (2023). Maximising Weather Forecasting Accuracy through the Utilisation of Graph Neural Networks and Dynamic GNNs. *arXiv preprint arXiv:2301.12471*.

23) Kong, Jian-Lei, Xiao-Meng Fan, Xue-Bo Jin, Ting-Li Su, Yu-Ting Bai, Hui-Jun Ma, and Min Zuo. "BMAE-Net: A data-driven weather prediction network for smart agriculture." *Agronomy* 13, no. 3 (2023): 625.

24) Han, Yan, Lihua Mi, Lian Shen, C. S. Cai, Yuchen Liu, Kai Li, and Guoji Xu. "A short-term wind speed prediction method utilizing novel hybrid deep learning algorithms to correct numerical weather forecasting." *Applied Energy* 312 (2022): 118777.

25) Michael, Neethu Elizabeth, Shazia Hasan, Ahmed Al-Durra, and Manohar Mishra. "Short-term solar irradiance forecasting based on a novel Bayesian optimized deep Long Short-Term Memory neural network." *Applied Energy* 324 (2022): 119727.

26) Weyn, J. A., Durran, D. R., & Caruana, R. (2020). Improving data-driven global weather prediction using deep convolutional neural networks on a cubed sphere. *Journal of Advances in Modeling Earth Systems*, *12*(9), e2020MS002109.

27) Scher, S., & Messori, G. (2021). Ensemble methods for neural network-based weather forecasts. *Journal of Advances in Modeling Earth Systems*, *13*(2).

28) Zhang, Zao, and Yuan Dong. "Temperature forecasting via convolutional recurrent neural networks based on time-series data." *Complexity* 2020 (2020): 1-8.

29) Kohli M & Arora S, Chaotic grey wolf optimization algorithm for constrained optimization problems, J Comput Des Eng, 5(4) (2018) 458–472, https://doi.org/10.1016/ j.jcde.2017.02.005.

30) W, Jin J, Wang B, Li K, Liang C, Dong J, Zhao S. Intelligence in Tourist Destinations Management: Improved Attention-based Gated Recurrent Unit Model for Accurate Tourist Flow Forecasting. *Sustainability*. 2020; 12(4):1390. https://doi.org/10.3390/su12041390

31) https://storage.googleapis.com/tensorflow/tf-keras-datasets/jena_climate_2009_2016.csv.zip