A novel approach to optimize LSTM Hyperparameter using the Chimp Optimization Algorithm for the Pressure Ventilator Prediction

Fatma Refaat Ahmed
University of Sharjah

Samira Ahmed Alsenany
Princess Nourah bint Abdulrahman University

Sally Mohammed Farghaly Abdelaliem (✉ sally.farghaly@alexu.edu.eg)
Princess Nourah bint Abdulrahman University

Mohanad A. Deif
Modern University of Technology and Information (MTI) University

Article

Keywords:

Posted Date: September 11th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3319590/v1

License: ☑️ 📧 This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Scientific Reports on November 27th, 2023. See the published version at https://doi.org/10.1038/s41598-023-47837-8.
A novel approach to optimize LSTM Hyperparameter using the Chimp Optimization Algorithm for the Pressure Ventilator Prediction

Fatma Refaat Ahmed 1, Samira Ahmed Alsenany 2, Sally Mohammed Farghaly Abdelaliem 3,* and Mohanad A. Deif 4

1 University of Sharjah, College of Health Sciences, Department of Nursing, Sharjah, UAE. Critical Care and Emergency Nursing Department, Faculty of Nursing, Alexandria University, Alexandria, Egypt; fahmed@sharjah.ac.ae
2 Department of Community Health Nursing, College of Nursing, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia; SAAISenany@pnu.edu.sa
3 Department of Nursing Management and Education, College of Nursing, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh 11671, Saudi Arabia Riyadh; smfarghaly@pnu.edu.sa; sally.farghaly@alexu.edu.eg
4 Department of Bioelectronics, Modern University of Technology and Information (MTI) University, Cairo 11728, Egypt; mohand.deif@eng.mti.edu.eg

* Correspondence: sally.farghaly@alexu.edu.eg / smfarghaly@pnu.edu.sa

Abstract:

Mechanical ventilation plays a vital role in the treatment of patients suffering from severe lung disease. In times of a pandemic, it becomes crucial to develop ventilators that can automatically adjust parameters during the treatment process. To address this need, a study was conducted to predict the pressure exerted on the patient by the ventilator. This prediction was based on various factors, including the ventilator's parameters and the patient's condition, using a specialized model known as Long Short-Term Memory (LSTM). In order to optimize the LSTM model and improve the accuracy of the prediction results, an algorithm called Chimp Optimization Algorithm (ChoA) was employed. The combination of LSTM and ChoA resulted in the creation of the LSTM-ChoA model, which effectively addressed the challenge of selecting appropriate hyperparameters for the LSTM model. Experimental findings demonstrated that the LSTM-ChoA model outperformed other optimization algorithms such as whale grey wolf optimizer (GWO), optimization algorithm (WOA), and particle swarm optimization (PSO), as well as regression models including K-nearest neighbor (KNN) Regressor, Random and Forest (RF) Regressor, and Support Vector Machine (SVM) Regressor, in terms of accurately predicting ventilator pressure.

1- Introduction

1.1 Background

Numerous illnesses call for a ventilator in medicine and medical transportation systems. The poliomyelitis pandemic in the 1950s provided conclusive evidence of a mechanical ventilator's effectiveness[1]–[3]. The ventilator can assist in moving air into and out of the lungs if a person has respiratory failure, hypoxemia, or hypercapnia and cannot breathe properly[4][5]. Furthermore, if someone has surgery under general anesthesia, they will also need a ventilator for adequate breathing. However, because mechanical ventilators are time-consuming, expensive, and ineffective, it is particularly challenging to have them available for all patients during a pandemic. Machine learning,
on the other hand, can support autonomous pressure selection and prediction[6]. Many traditional mechanical ventilator prediction techniques have been introduced in recent years. Nonetheless, they should have paid more attention to the costly newly developed technologies and correct pressure because a doctor manually adjusted them [7].

Combining derivative, integral, and proportional controls is the industry standard for PID controllers. In this fusion, the discrepancy from the desired waveform is employed as input, leading to adjustments in the pressure chamber to rectify the output waveform and minimize the margin of deviation. Pressure inaccuracies emerge from this operating design, which calls for continual human surveillance and intervention by medical experts. This method complies with the required requirements and safety standards; however it lacks the adaptability and quickness needed for a variety of clinical scenarios. Therefore, this problem might be solved with a dynamic controller that can continuously adjust pressure settings. This is the precise situation where replacing the traditional PID controller algorithm with machine learning comes into play.

Due to the recent COVID-19 pandemic, a considerable array of open-source ventilator designs has emerged. This situation has paved the way for numerous endeavors to enhance these ventilators using advanced techniques and systems. These efforts involve incorporating decision support methods reliant on machine learning models that are finely tuned to incorporate patient responses. The overarching aim is to enhance the accuracy of pressure values.

There have been more and more studies on the forecasting of ventilator pressure using LSTM networks because it has been shown that the LSTM, as a type of Recurrent Neural Network (RNN)[8][9], is suited for evaluating long-series data. The relevant hyperparameters must be changed to build an LSTM network model that satisfies the specifications[10], and researchers frequently choose these values based on past knowledge and experience[11]. It could be necessary to adjust the pertinent parameters for specific problems manually[12] [13].

### 1.2 Related works and problem statement

Numerous studies have helped to advance the LSTM network optimization techniques in recent years. Table 1 provides information about the models that were used as well as a list of current related research.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Optimization model</th>
<th>Research aim</th>
</tr>
</thead>
<tbody>
<tr>
<td>[14]</td>
<td>Gradient clipping and weight initialization</td>
<td>Address the challenges of training LSTMs by proposing techniques to tackle the vanishing/exploding gradient problem.</td>
</tr>
<tr>
<td>[15]</td>
<td>Proper weight initialization</td>
<td>The paper discussed the importance of appropriate weight initialization techniques in facilitating the training of deep neural networks, including LSTMs.</td>
</tr>
<tr>
<td>[16]</td>
<td>Non-saturating activation functions and gradient clipping</td>
<td>The study explored the challenges associated with training RNNs, including LSTMs, and proposed techniques to address the vanishing/exploding gradient problem.</td>
</tr>
<tr>
<td>[17]</td>
<td>Adam optimization algorithm.</td>
<td>The authors introduced Adam as an efficient and effective optimization algorithm for training neural networks, including LSTM networks.</td>
</tr>
</tbody>
</table>
LSTMs, and demonstrated its benefits in handling different optimization problems.

[18] Scheduled sampling
The study introduced scheduled sampling as a technique for training sequence prediction models, including LSTMs. It addressed the discrepancy between training and inference by gradually replacing ground truth inputs with model predictions during training.

[19] Layer normalization
The internal covariate shift issue in deep neural networks, including LSTMs, was discussed in the study, and layer normalization was suggested as a solution. It sought to expedite training and stabilize optimization.

The research introduced the residual learning framework, which allows for training very deep neural networks, including LSTMs, by using skip connections. It mitigated the vanishing/exploding gradient problem and enabled the training of deeper LSTM architectures.

[21] Stochastic gradient descent (SGD), mini-batch training, learning rate schedules, and adaptive optimization algorithms
The study provided an overview of optimization methods for large-scale machine learning tasks, including LSTM optimization. It discussed various techniques to optimize LSTMs and other models in the context of large-scale datasets and computational constraints.

[22] the modified bald eagle search (MBES) algorithm, PSO, GA, GWO
the select the LSTM hyperparameters using

Every optimization technique comes with its own set of advantages and disadvantages. The selection of an approach hinges on the particular demands of the task, the network’s structure, and the computational resources at hand. It is often necessary to experiment and fine-tune these techniques to achieve optimal results for a given LSTM-based application.

1. Gradient Clipping and Weight Initialization:
   - Benefits: Gradient clipping helps address the vanishing/exploding gradient problem by imposing a threshold on the gradients during training. It prevents gradients from becoming too large or too small, leading to more stable and effective training. Proper weight initialization techniques, such as Glorot and He initialization, ensure that the weights are initialized in a way that promotes efficient gradient flow and convergence.
   - Drawbacks: Gradient clipping can potentially introduce biases to the gradient updates and may require careful tuning of the threshold value. Weight initialization techniques may not always guarantee the optimal initialization for all network architectures or tasks, and finding the appropriate initialization scheme can still be a trial-and-error process.

2. Non-saturating Activation Functions and Gradient Clipping:
   - Benefits: Non-saturating activation functions, such as ReLU (Rectified Linear Unit), help mitigate the vanishing gradient problem by avoiding saturation in the activation values. They facilitate the flow of gradients and enable better learning in deep architectures. Gradient clipping, as mentioned earlier, prevents gradients from becoming too large, ensuring more stable training.
   - Drawbacks: Activation functions such as ReLU, which do not saturate, can encounter the issue of the "dying ReLU" phenomenon. This occurs when certain neurons become inactive and fail to recuperate throughout the training process. This issue can hinder the learning process and network performance. Gradient clipping, if applied
too aggressively, can result in underutilization of the gradient information and slow convergence.

3. Adam Optimization:
   - Benefits: Adam combines adaptive learning rates and momentum to efficiently optimize LSTM networks. By adjusting the learning rate on a per-parameter basis, it achieves enhanced optimization performance and quicker convergence. Adam also maintains separate learning rates for each parameter, making it suitable for dealing with sparse gradients and noisy data.
   - Drawbacks: Adam has several hyperparameters that require careful tuning, such as the learning rate, momentum parameters, and exponential decay rates. Poor hyperparameter selection can lead to suboptimal results or difficulties in convergence. Additionally, Adam may not always generalize well to all types of optimization problems and could benefit from modifications in certain scenarios.

4. Scheduled Sampling:
   - Benefits: Scheduled sampling addresses the discrepancy between training and inference by gradually introducing model predictions during training. It allows the model to adapt to the errors made by its own predictions, leading to improved performance during inference. This technique is particularly useful in sequence prediction tasks, where the model needs to generate accurate outputs based on its own predictions.
   - Drawbacks: Scheduled sampling introduces a discrepancy between the training and inference processes, which can make the optimization task more challenging. Determining the appropriate schedule for introducing model predictions requires careful consideration and may vary depending on the specific task and dataset.

5. Layer Normalization:
   - Benefits: Layer normalization addresses the internal covariate shift problem by normalizing the inputs to each layer. It helps stabilize the optimization process, accelerates training, and allows for faster convergence. Layer normalization also provides better generalization and performance, particularly in deep networks.
   - Drawbacks: Layer normalization introduces additional computational overhead, as it requires calculating the mean and variance of each layer's inputs. This overhead can make training slower compared to other normalization techniques. Additionally, the benefits of layer normalization may vary depending on the specific architecture and task, and it might not be universally applicable in all cases.

6. Residual Learning:
   - Benefits: Residual learning, achieved through skip connections, allows for training very deep networks, including LSTMs. It mitigates the vanishing/exploding gradient problem by providing shortcut connections that facilitate gradient flow. Residual connections enable the training of deeper LSTM architectures, leading to improved accuracy and better utilization of network depth.
   - Drawbacks: The main challenge with residual learning is designing appropriate skip connections and ensuring that the information flow through the network is optimized. Poorly designed skip connections or overly deep architectures can still suffer from optimization difficulties. Additionally, residual learning may require careful tuning of hyperparameters and architectural decisions.
1.3 Motivations, contributions, and article organization

In order to address these challenges in our study, the parameters of the LSTM model are fine-tuned using the ChoA algorithm, resulting in the development of the LSTM-ChoA prediction model. The experimental findings demonstrate that this optimized model effectively predicts ventilator pressure for patients, enabling mechanical ventilators to provide tailored patient care. The key contributions of this research can be summarized as follows:

(1) The ventilator dataset was explored, analyzed, and added new features.
(2) The ChoA algorithm optimized the parameters of the LSTM, and the ChoA-LSTM forecasting model was subsequently constructed.
(3) ChoA algorithm's efficiency was compared with other metaheuristic optimization algorithms.
(4) The prediction efficiency of LSTM-ChoA was compared with other regressor models.

This research article comprises several sections, namely: related works, inspiration, study objectives, theoretical framework and approach, experimentation, experimental results, and conclusion. The theoretical and methodological aspects encompass two key components: an explanation of the LSTM model theory, the ChoA algorithm, and a concise overview of our proposed methodology.

2- Materials and Methods
2.1 Materials

The ventilator data used in this work was obtained from the Google Brain public dataset, which is publicly accessible on Kaggle [23]. An artificial bellows test lung was attached to a custom open-source ventilator using a respiratory circuit to create the dataset. Figure 2 shows the configuration, with the predicted state variable (airway pressure) indicated in blue and the two control inputs highlighted in green. The inspiratory solenoid valve's percentage of opening, which allows air into the lung, is represented by the first control input. It has a scale of 0 to 100, with 0 signifying total closure (no air allowed in) and 100 signifying complete opening. A binary variable showing whether the exploratory valve is open (1) or closed (0), allowing air to be discharged, serves as the second control input. The goal of positive end-expiratory pressure (PEEP) is to keep the lungs from collapsing [24].

![Figure 1: The diagram of the respiratory circuit](image-url)
Each time series corresponds to a breath that lasts about three seconds. About 75,000 for training and 50,000 test breaths comprise the dataset's five features. Information on significant features and intended columns is compiled in Table 2.

<table>
<thead>
<tr>
<th>Feature Name</th>
<th>Feature Abbreviation (Unit)</th>
<th>Feature Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resistance</td>
<td>R (cmH2O/L/S)</td>
<td>restriction of the airway</td>
</tr>
<tr>
<td>Compliance</td>
<td>C (mL/cmH2O)</td>
<td>The ability of the lungs to stretch and expand</td>
</tr>
<tr>
<td>Time-step</td>
<td>Time-step</td>
<td>The actual time stamps.</td>
</tr>
<tr>
<td>Inspiratory Solenoid Valve</td>
<td>ISV</td>
<td>Manipulation command for the inhalation solenoid valve</td>
</tr>
<tr>
<td>Exploratory Solenoid Valve</td>
<td>ESV</td>
<td>The input utilized for controlling the exploratory solenoid valve.</td>
</tr>
<tr>
<td>Airway pressure</td>
<td>P (cmH2O)</td>
<td>The recorded airway pressure within the respiratory circuit is displayed in the designated column.</td>
</tr>
</tbody>
</table>

Among the observed values, R equal to 20 and C were found to be the most common in patients with severe lung disease[10]. The extent of openness of the inspiratory solenoid valve (ISV) is quantified as a continuous parameter spanning from 0 to 100. Here, 0 signifies a fully closed position, while 100 corresponds to complete openness. Alternatively, the exploratory valve status (ESV) serves as a binary variable denoting whether the valve is in the open state (1) or the closed state (0), facilitating air release.

2.2 Preliminaries

2.2.1 Long Short-Term Memory (LSTM)

The construction of the LSTM, a particular kind of RNN [25], [26], is depicted in Figure 1. Its hidden layer comprises one or more memory cells, each with an input, output, and forget gate [27], [28].
Figure 2: LSTM model structure

- **The forget gate** ($f_t$): is responsible for deciding whether to retain the long-term state $c$ and is computed using the current time step $t$ input $x_t$ and the output $h_{t−1}$ from the preceding time step $t−1$. The corresponding formula for this operation is:

$$f_t = \sigma(W_f \times [h_{t-1}, x_t] + B_f)$$  \hspace{1cm} (1)

The matrix $W_f$ corresponds to the weights linked with the forget gate, while $B_f$ denotes the biased term. The sigmoid function $\sigma(\cdot)$ is applied to the combined inputs to compute the forget gate's decision.

- **The input gate's** is responsible for generating a new candidate cell state $\tilde{c}$, performing relevant computations, and controlling the amount of information to be added. The calculation formula for the input gate is as follows:

$$i_t = \sigma(W_i \times [h_{t-1}, x_t] + B_i)$$  \hspace{1cm} (2)

$$\tilde{c}_t = \tanh(W_c \times [h_{t-1}, x_t] + B_c)$$  \hspace{1cm} (3)

In the given equation, the current input cell state represented by "$i_t$", while $W_c$ corresponds to the weight matrix linked to the cell state, and $B_c$ signifies the bias term associated with the cell state. Similarly, $i_t$ represents the output generated by the input gate, $W_i$ is indicative of the weight matrix for the input gate, and $B_i$ stands for the bias term pertaining to the input gate. The matrix denoted as $[h_{t-1}, x_t]$ encompasses two vectors: $x_t$, which signifies the input at the present time step ($t$), and $h_{t-1}$, representing the output from the prior time step ($t-1$). The sigmoid activation function is denoted by $\sigma(\cdot)$, while the hyperbolic tangent function is represented as $\tanh(\cdot)$. 
Output gate: produces the output \( f_t \) using a sigmoid function \( \sigma(\cdot) \) that takes the input \( x_t \) at the current time step \( (t) \) and the output \( h_{t-1} \) from the previous time step \((t - 1)\) as inputs. The calculation can be expressed as follows:

\[
O_t = \sigma(W_o \times [h_{t-1}, x_t] + B_o)
\]

\[
c_t = f_t \times c_{t-1} + i_t \circ \hat{c}_t
\]

\[
h_t = o_t \times \tanh(c_t)
\]

\( B_o \) represents the output gate's bias term, and \( W_o \) is the weight of the output gates.

The LSTM structure, with its unique three-gate architecture and hidden state with memory capabilities, effectively addresses the issue of long-term dependencies by capturing long-term historical information. The hidden state undergoes a series of operations. Firstly, the forget gate \( C_t \) at the current time step \((t)\) regulates which information from the previous hidden state \( C_{t-1} \) at time step \( t - 1 \) needs to be discarded and which information should be retained. Secondly, the structure of the hidden state selectively discards certain data while utilizing the input gate to incorporate new information in conjunction with the forget gate. Finally, the cell state \( C_t \) is updated through a sequence of computations. The LSTM employs the output gate, the cell state \( C_t \), and a tanh layer to calculate the final output value \( h_t \).

2.2.2 Chimp Optimization Algorithm (ChoA)

This model draws inspiration from chimpanzee intelligence and breeding behavior in cluster hunting [29]–[31]. Figure 3 depicts attack, chase, barrier, and driver as the four methods used to replicate the attitude. In (7) and (8), the scientific equations for the provided approach's driver and attacker are mentioned[32].

\[
D = |C \cdot \alpha_{prey} - m \cdot \alpha_{chimp}|
\]

\[
\alpha_{chimp}(n + 1) = |\alpha_{prey} - a \cdot d|
\]

Where the co-efficient vectors are \( m, c, \) and \( d, \) these elements can be justified by (9) through (11).

\[
a = 2 \cdot l \cdot r_1 - 1
\]

\[
c = 2 \cdot r_2
\]

\[
m = \text{chaotic value}
\]

Where \( l \) is a constant that drops along a line from 2.5 to 0 throughout the iterations [33], \( r_1 \) and \( r_2 \) are random values between 0 and 1, and \( m \) is a disordered vector. The attacker, barrier, chaser, and driver are the best outputs with optimal capabilities for mathematically simulating this system[34], [35]. The information provided by the top four outcomes will help the remaining residents upgrade their location. Equations (12), (13), (14), and (15) provide their statistical proof, respectively[36].

\[
d_{\text{attack}} = |C_1 \cdot \alpha_{\text{attacker}} - m_1 \cdot x_n|
\]

\[
d_{\text{barrier}} = |C_2 \cdot \alpha_{\text{barrier}} - m_2 \cdot x_n|
\]

\[
d_{\text{chaser}} = |C_3 \cdot \alpha_{\text{chaser}} - m_3 \cdot x_n|
\]

\[
d_{\text{driver}} = |C_4 \cdot \alpha_{\text{driver}} - m_4 \cdot x_n|
\]
Following this, the chimpanzees’ next point is reorganized using (16) through (19):

\[ x_1 = \alpha_{\text{attacker}} - a_1 \cdot d_{\text{attacker}} \]  
\[ x_2 = \alpha_{\text{barrier}} - a_2 \cdot d_{\text{barrier}} \]  
\[ x_3 = \alpha_{\text{chaser}} - a_3 \cdot d_{\text{chaser}} \]  
\[ x_4 = \alpha_{\text{driver}} - a_4 \cdot d_{\text{driver}} \]  

The positions are upgraded by using Equation (20):

\[ x_1 = \frac{x_1 + x_2 + x_3 + x_4}{4} \]  

Then, Equation (21) is implemented once the positions have been upgraded.

\[ \alpha_{\text{chimp}} (n + 1) = \begin{cases} 
\alpha_{\text{prey}} - x \cdot d , & \emptyset < 0.5 \\
\text{chaotic} , & \emptyset < 0.5 
\end{cases} \]  

--Figure 3. Updating the ChoA position.

### 2.3 Methodology

As shown in Figure 4, our proposed approach for ventilator pressure prediction includes five steps. The study encompassed several stages, starting with Exploratory Data Analysis (EDA) and progressing through feature extraction, the development of a regression model, an assessment of the model’s performance, and a subsequent comparison of the outcomes achieved by the proposed regression model with those of benchmark models. Each phase is described in detail below.
2.3.1 Exploratory Data Analysis

This process enabled us to extract additional meaningful insights from the ventilator dataset, [36] uncover distinct patterns within the data, and identify the most suitable models that utilize the available features to achieve faster convergence [39] and minimize prediction errors [40].

The exploratory Data Analysis involves studying the correlation between dataset variables and airway pressure patterns during the inhalation and exhalation phases. Finally, analysis of the Effects R-C pairs have on pressure distribution. Our study uses Spearman's rank correlation ($\rho$) to assess the correlation between dataset features as follows [41]:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

Where $n$ is the number of observations and $d_i$ is the difference between each observation.

2.3.2 Extracted features

Based on previous analysis, it was decided to create and extract four features that improve the model predictions and converge the model faster [42]. The following features were extracted:

1) Lung setting ($\delta$) = R*C: It indicates the change in volume per change in flow
2) CISV is defined as the cumulative sum of the control input for the inspiratory solenoid valve.
3) Integral is the difference between consecutive time steps.
4) Differential = indicates the difference of consecutive samples.
2.3.3 Development Regression Algorithm

The process of selecting a set of hyperparameters that yield optimal performance for the Long Short-Term Memory (LSTM) model on the given data is referred to as Hyperparameter Optimization (HPO). In this work, the Chimp Optimization Algorithm (ChoA) is employed to determine the best values for the LSTM Model Hyperparameters. These hyperparameters include the Number of Iterations (Ni), Learning Rate (La), Hidden Units (NHU), and Number of Layers (NL). Figure 5 portrays the flow chart that presents the LSTM forecasting model known as the LSTM-ChoA model. The process encompasses distinct steps outlined below:

**Figure 5:** Flowchart of the LSTM-ChoA algorithm

**Step (1):** Initialize LSTM \{E, La, NHU, NL\} parameters and the ChoA algorithm parameters that include the chimp population \{r_1, r_2, l\} and the constants \{a, c, m\}.  

**Step (2):** The mean square error gained from training the LSTM network was used to compute the fitness function, which corresponds to the cost function value. This fitness value was continuously updated in real-time as the Chimp optimization algorithm progressed. Within the range of iterations, Equation (9) was utilized to determine the position of each chimp, and subsequently, the chimps were randomly divided into independent groups.
The cost function is Mean Squared Error (MSE), which distinguishes between real and predicted data. The MSE function must be reduced during LSTM training by updating hyperparameter values. The MSE is presented in Equation (23).

\[
\text{MSE} = \frac{\sum_{i=0}^{n} (y - y')^2}{n}
\]  

(23)

**Step (4):** Assign the best chimp in each group using Equations (12), (13), (14), and (15), respectively.

**Step (3):** If the current new position was better, the old position was updated using Equation (20).

**Step (5):** The LSTM network was trained based on the optimal parameter combination, and testing samples were utilized for forecasting.

### 2.3.4 Benchmark models

The evaluation of ChoA’s performance involved a comparison with several benchmark optimization methods, such as whale optimization algorithm (WOA), grey wolf optimizer (GWO) [44], and particle swarm optimization (PSO) [43]. The initial parameter values for each optimizer can be found in Table 3.

Furthermore, the LSTM-ChoA algorithm has been compared with regression models as baselines: Random forest (RF) Regressor, Support Vector Machine (SVM) Regressor, and K-nearest neighbor (KNN) Regressor. The optimal Hyper-parameter for baseline models was selected using grid search in sklearn Libraries [37], [38]. In Table 4, the search space for the hyperparameters of the regressor models is presented, along with the selected hyperparameters obtained through grid search.

### Table 3: Parameter value of PSO, GWO, WOA, and ChoA.

<table>
<thead>
<tr>
<th>Optimizer</th>
<th>Hyper-parameter name</th>
<th>Parameter value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Common setting</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Population size</td>
<td>P = 10</td>
<td></td>
</tr>
<tr>
<td>Maximum iteration</td>
<td>N = 200</td>
<td></td>
</tr>
<tr>
<td>Runs</td>
<td>r = 20</td>
<td></td>
</tr>
<tr>
<td>Probability</td>
<td>Pr = 0.5</td>
<td></td>
</tr>
<tr>
<td><strong>PSO</strong></td>
<td>Number of particles</td>
<td>S=30</td>
</tr>
<tr>
<td>Cognitive factor and social factor.</td>
<td>C1=C2 =0.75</td>
<td></td>
</tr>
<tr>
<td><strong>GWO</strong></td>
<td>Population size</td>
<td>P=10</td>
</tr>
<tr>
<td>gene mutation</td>
<td>Gm= 0.10</td>
<td></td>
</tr>
<tr>
<td>gene crossover</td>
<td>Gc= 0.5</td>
<td></td>
</tr>
<tr>
<td>generations number</td>
<td>Gn= 1.5</td>
<td></td>
</tr>
<tr>
<td><strong>WOA</strong></td>
<td>Whales number</td>
<td>Whales number</td>
</tr>
<tr>
<td><strong>ChoA</strong></td>
<td>Random vectors</td>
<td>in the range of [0, 1]</td>
</tr>
<tr>
<td>constant</td>
<td>( t = 2.5 )</td>
<td></td>
</tr>
</tbody>
</table>
Table 4: The range of potential values for the hyperparameters of the regressor models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Hyper-parameter</th>
<th>Search Space</th>
<th>Selected Hyper-parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>n estimators</td>
<td>[10,100]</td>
<td>36</td>
</tr>
<tr>
<td>RF</td>
<td>max depth</td>
<td>[5,50]</td>
<td>6</td>
</tr>
<tr>
<td>min samples leaf</td>
<td>[1,11]</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>min samples split</td>
<td>[2,11]</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>criterion</td>
<td>['mse', 'mae']</td>
<td>mse</td>
</tr>
<tr>
<td></td>
<td>max features</td>
<td>[1,13]</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>[0.1,50]</td>
<td>50</td>
</tr>
<tr>
<td>SVM</td>
<td>kernel</td>
<td>['linear', 'poly', 'rbf', 'sigmoid']</td>
<td>rbf</td>
</tr>
<tr>
<td></td>
<td>epsilon</td>
<td>[0.001,1]</td>
<td>[0.001]</td>
</tr>
<tr>
<td>KNN</td>
<td>n neighbors</td>
<td>[1,20]</td>
<td>5</td>
</tr>
</tbody>
</table>

2.1 Implementation

The methodology introduced was put into practice using Jupyter Notebook version 6.5.4 [45] along with Python version 3.6.7 [46]. The execution took place on a personal computer equipped with an Intel Core i5 processor, 4 GB of RAM, and the Windows 10 Professional 64-bit operating system [47].

3- Results and discussion

This section introduces Exploratory Data Analysis for the ventilator dataset we used and the LSTM- ChoA model results. A complete model evaluation and comparison are presented in Section 2.3.

Figure 6 displays the correlation matrix, illustrating the correlation coefficients among the different elements of interest. A value of -1 denotes a high negative correlation, while a value of 1 denotes a strong positive correlation, and the matrix determines the linear correlation between variables. The diagonal values represent the dependence of each variable on itself, also known as autocorrelation.
The correlation matrix of features indicates that there is a relatively stronger correlation between airway pressure and both features (ISV and Time step) compared to the other variables. It can be explained through the principle of the ventilator. When the value of ISV changes, the air pressure will change in the patient’s bronchi, corresponding to the value of the feature "Pressure" in the dataset.

Besides airway pressure, property C also has a degree of dependency significantly with ISV. When we increase or decrease the value of the inspiratory electronic valve, the degree of lung expansion will also change (varying with the volume of air transmitted into the lungs). The remaining features have a low correlation with each other.

Figure 7 shows the pressure magnitude distributions when the expiratory valve is closed and open. From this figure, we can see that when the expiratory valve is open, the pressure distribution is left-skewed, meaning that the airway pressure lies in the lower ranges in such a case. On the other hand, when the valve is closed (referred to as the inhalation phase), the pressure lies in a wider range, which is expected as pressure variations occur during inhalation. Furthermore, notice the abrupt peaks in the distribution when the expiratory valve is closed (during inhalation), which produces small peaks in the overall distribution. The reason for these small peaks represents a slight EIV valve leak.
To illustrate how the airway pressure changes within a complete breath cycle. Figure 8 shows the airway pressure and input valve positions (opened/closed) for four different breaths. Each breath represents 80 recorded time steps.
During a breath cycle, when the expiratory valve is closed (inhalation), the pressure gradually increases as the inspiratory valve open percentage increases. Interesting to note, however, is that there is a certain ‘delay time’ between the change of valve position (control variable) and the pressure (response). Once the pressure reaches the peak value, the expiratory valve is opened as the exhalation phase begins. The pressure decreases rapidly until it reaches an asymptote. This cycle repeats continuously. It understands from this that we realize that pressure at consecutive time steps bears strong correlations with each other, meaning that sequential treatment of the data can prove advantageous.

Figure 9 shows the effects of R-C pairs on airway pressure distribution. It has nine combinations of R and C. It has nine varieties of experiments; C can be 10, 20, or 50, and R can be 5, 20, or 50. It is observed that, with different R-C pairs, the pressure distribution varies for the output valve closed and open. Therefore the adding R-C pair as a new feature in the training dataset may have a positive impact on the LSTM-ChoA model's ability to predict ventilator pressure.
This study has extracted four features (δ, CISV, Integral, and Differential) to improve the model predictions. Figure 10 shows the airway pressure distribution based on extracted features and nine R-C combinations.
Figure 10: Airway pressure distribution based on extracted features (a) $\delta$, (b) CISV, (c) Integral, and (d) Differential

It is observed in Figure 10. a, b, and c that the air pressure variation will appear and certain distinct clusters are formed. At the same time, in Figure 10.d, the differential feature fails to discriminate our dataset into certain clusters. Therefore, the features ($\delta$, CISV, and Integral) will be preserved while will Differential feature will be neglected.

For the prediction of airway pressure, two experiments were conducted in this study; in the first experiment, four metaheuristic optimization models (PSO, GWO, WOA, and ChoA) were applied to find an efficient LSTM hyperparameter ($N_i$, $L_R$, $N_{HU}$, and $N_l$) can better improve the model accuracy. In the second experiment, the prediction efficiency of LSTM-ChoA was compared with other regressor models. In two experiments, the training set accounts for 80% of the entire ventilator dataset and is used to establish regressor models. The remaining 20% is the test set used to measure the selected hyperparameters' performance.

Table 5 showcases the parameter values associated with LSTM and the corresponding computed errors derived from the initial experiment. The outcomes clearly illustrate that optimizing hyperparameters using ChoA can substantially enhance the regression performance of the LSTM model. This improvement is generally evident. ChoA outperformed Another Comparative
optimization techniques PSO, GWO, and WOA, with the lowest MSE value (0.0852). The LSTM-PSO and LSTM-WOA have a similar value of MSE.

**Table 5:** The LSTM hyper-parameters value and prediction errors based on optimization algorithms

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>hyper-parameters value</th>
<th>MSE</th>
<th>CT (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM-ChoA</td>
<td>95 0.0852 93 25</td>
<td>0.0852</td>
<td>23.21</td>
</tr>
<tr>
<td>LSTM-PSO</td>
<td>86 0.2192 54 78</td>
<td>0.2192</td>
<td>13.40</td>
</tr>
<tr>
<td>LSTM-GWO</td>
<td>43 0.1789 96 51</td>
<td>0.1789</td>
<td>18.12</td>
</tr>
<tr>
<td>LSTM-WOA</td>
<td>23 0.2616 61 95</td>
<td>0.2616</td>
<td>8.58</td>
</tr>
</tbody>
</table>

Furthermore, the overall time required (referred to as computational time or CT) to accomplish a hyperparameter optimization procedure using 5-fold cross-validation is employed as a metric for assessing model efficiency. It has been observed that the computational time of ChoA is generally lengthier compared to other optimization methods, whereas WOA consistently achieves the shortest time.

Figures 11 depict the convergence curves of the LSTM model employing four distinct optimization algorithms. These figures reveal that the utilization of ChoA resulted in superior performance for the LSTM model. Specifically, the LSTM-ChoA model demonstrated better outcomes compared to other regression models, exhibiting faster and deeper convergence towards the optimal solution. Conversely, the LSTM-GWO approach also exhibited competitive performance in this study.

![Figure 11: Convergence curves of the LSTM regression model based on (a) ChoA, (b) PSO, (c) WOA and (d) GWO](image-url)
Considering that the regression outcome of each model is the average mean squared error (MSE) derived from 10 independent iterations, a logical step was to employ a two-sample t-test at a 95% confidence level. This test aimed to determine whether the classification performance achieved by the proposed LSTM-ChoA method exhibited a noteworthy superiority over other approaches. In this statistical evaluation, the null hypothesis suggested that the classification performance of distinct methods was comparable. Should the calculated p-value fall below 0.05, it would signify the rejection of the null hypothesis, indicating a significant divergence in classification performance between the methods being compared.

The outcomes of the t-test, along with corresponding p-values, are presented in Table 6, with LSTM-ChoA used as the benchmark method. It is evident from the results that the regression performance of LSTM-ChoA displayed a notable improvement (p-value < 0.05) across the 10 independent iterations.

<table>
<thead>
<tr>
<th>Run number</th>
<th>P-value</th>
<th>LSTM-PSO</th>
<th>LSTM-GWO</th>
<th>LSTM-WOA</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0549</td>
<td>0.0231</td>
<td>0.0426</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.0350</td>
<td>0.0354</td>
<td>0.0152</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.0131</td>
<td>0.0392</td>
<td>0.0247</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.0264</td>
<td>0.0513</td>
<td>0.0221</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.0380</td>
<td>0.0125</td>
<td>0.0537</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.0123</td>
<td>0.0512</td>
<td>0.0442</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.0547</td>
<td>0.0128</td>
<td>0.0161</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.0441</td>
<td>0.0463</td>
<td>0.0334</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.0305</td>
<td>0.0130</td>
<td>0.0270</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.0154</td>
<td>0.0118</td>
<td>0.0288</td>
<td></td>
</tr>
</tbody>
</table>

In order to demonstrate the superior performance of the LSTM-ChoA prediction model, it was compared to other regression models such as RF, SVM, and KNN in the second experiment. Table 7 presents a summary of the results obtained from these predictive models in terms of mean squared error (MSE). As mentioned earlier, the hyperparameters for these models were determined using a grid search. It is evident from the table that the LSTM-ChoA model outperforms the other regression models by achieving the lowest MSE, highlighting its effectiveness in making accurate predictions.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>0.4161</td>
</tr>
<tr>
<td>SVM</td>
<td>0.2362</td>
</tr>
<tr>
<td>KNN</td>
<td>0.3142</td>
</tr>
<tr>
<td>LSTM-ChoA</td>
<td>0.0852</td>
</tr>
</tbody>
</table>
Figure 12 displays a contrast between the factual values and the projected values for airway pressure. The figure unmistakably reveals that the prediction line generated by the RF model notably diverges from the actual pressure line, underscoring its subpar predictive performance. On the other hand, the prediction lines produced by the LSTM-ChoA model closely align with the actual lines, indicating its ability to accurately capture the underlying patterns and trends in the data.

![Figure 12: Figure 7. The prediction results of (a) LSTM-ChoA, (b) SVM, (c) KNN, and (d) RF](image)

The LSTM model built on ChoA provides a lot of benefits. The pressure ventilator prediction forecasting accuracy significantly increased as compared to the other models. You can sum up the LSTM-ChoA model as follows:

1. The Chimp Optimization Algorithm played a vital role in optimizing the hyperparameters of the LSTM model, leading to a substantial enhancement in forecasting accuracy. This improvement is evident from the results presented in Table 4. The MSE has decreased by about 14.8% when using ChoA compared with GWO, while the MSE has decreased by about 60% ChoA compared with PSO and WOA.

2. The results in Figures 11 have shown that both the LSTM-ChoA algorithms have the best convergence curve followed by LSTM-GWO and LSTM-PSO, LSTM-WOA at the end.
The t-test results with p-values in Table 5 show that the regression performance of LSTM-ChoA was significantly better (p-value < 0.05) on the ten independent runs.

The assessment of the proposed algorithm encompassed various benchmark methods including RF, SVM, and KNN. It becomes evident upon reviewing Table 6 and Figures 12 that the LSTM-ChoA model surpasses alternative regression models, attaining the lowest mean squared error (MSE).

Conclusion:

In this research paper, we made use of the ventilator pressure prediction dataset sourced from Google Brain on Kaggle. This dataset originates from an open-source ventilation system created by researchers at Princeton Lab. Our proposed approach for predicting airway pressure in ventilator respiratory circuits involves a comprehensive analysis. This entails exploratory data analysis, feature extraction, regression model creation, performance analysis, and comparison of the outcomes with reference models.

We discovered a significant relationship between airway pressure and parameters like ISV and Time step during the exploratory data analysis. The distribution of feature C also showed distinct peaks when the expiratory valve was closed, indicating a minor leak in the EIV valve, and feature C showed a considerable dependence on ISV. We retrieved four features (, CISV, Integral, and Differential) to improve the prediction model. This investigation strengthens the case for sequential data treatment's efficacy.

The LSTM Model Hyperparameters (NI, LR, NHU, and NL) were optimized using the Chimp Optimization Algorithm (ChoA). ChoA's effectiveness was assessed and contrasted with that of other benchmark optimization algorithms like PSO, GWO, and WOA. Moreover, the LSTM-ChoA model was compared to regression models (RF, SVM, and KNN) as baseline models. Through extensive experimentation, ChoA demonstrated superior performance in finding the optimal LSTM Model Hyperparameters. Consequently, the LSTM-ChoA model exhibited remarkable prediction capacity and achieved the highest accuracy among all the tested models. Hence, it can be deduced that the suggested LSTM-ChoA model surpasses alternative models in terms of prediction accuracy.

Author Contributions: Formulation, Fatma R. Ahmed. and Mohanad A. Deif.; methodology, Mohanad A. Deif and Sally M. Farghaly.; software, Mohanad A. Deif.; validation and investigation, Sally M. Farghaly and Fatma R. Ahmed.; formal analysis, Samira A. Alsenany and Sally M. Farghaly.; resources, Fatma R. Ahmed and Sally M. Farghaly.; preparing original draft, Mohanad A. Deif. and Sally M. Farghaly.; review and editing of writing, Samira A. Alsenany. and Sally M. Farghaly.; supervision, Samira A. Alsenan.; funding and visualization, Samira A. Alsenany. All authors have read and agreed to the published version of the manuscript.

Funding: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R279), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.
Conflicts of Interest: The authors have no conflict of interest to disclose

Data Availability Statement: The datasets analyzed during the current study are available on the Kaggle website, [https://www.kaggle.com/competitions/ventilator-pressure-prediction/data]

References:


Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Dataset.zip