Whale Optimization Algorithm based on Tent Chaotic Map for Feature Selection in Soft Sensors

Mothena Fakhri Shaker AlRijeb
Advanced Lightning, Power, and Energy Research (ALPER), Department of Electrical and Electronic Engineering, Faculty of Engineering, Universiti Putra Malaysia (UPM), Serdang, Malaysia | Faculty of Engineering, Aliraqia University, Baghdad, Iraq
mothena.f@aliraqia.edu.iq (corresponding author)

Mohammad Lutfi Othman
Department of Electrical and Electronic Engineering, Faculty of Engineering, Universiti Putra Malaysia (UPM), Serdang, Selangor, Malaysia
lutfi@upm.edu.my

Aris Ishak
Department of Electrical and Electronic Engineering, Faculty of Engineering, Universiti Putra Malaysia (UPM), Serdang, Selangor, Malaysia
ishak_ar@upm.edu.my

Mohd Khair Hassan
Department of Electrical and Electronic Engineering, Faculty of Engineering, Universiti Putra Malaysia (UPM), Serdang, Selangor, Malaysia
khair@upm.edu.my

Baraa Munqith Albaker
Faculty of Engineering, Aliraqia University, Baghdad, Iraq
baraamalbaker@ymail.com

Received: 17 March 2025 | Revised: 6 April 2025 and 18 April 2025 | Accepted: 22 April 2025

 DOI: https://doi.org/10.48084/etasr.10965

ABSTRACT

Irrelevant features in data collected from oil refineries affect system performance due to conflicts between normal data and detected faults. Selecting the relevant features from the data leads to better classification results. Optimization algorithms are successfully applied in the feature selection task in many systems. One of the powerful optimization algorithms that is used for feature selection is the Whale Optimization Algorithm (WOA), which is a nature-inspired metaheuristic optimization algorithm that mimics the social behavior of humpback whales. This study presents an improvement to WOA using a tent chaotic map to select the most relevant features and enhance performance. The Tent map mainly applies randomness to generate diversification into the search process and escape from local optima in WOA. The tent map is used for generating the initial population, producing values in control parameters, and updating the position of search agents. The proposed approach combines the tent map with WOA, called TWOA, to enrich population diversity, prevent premature convergence, and speed up convergence. TWOA is applied in a soft sensor with actual data collected from the Salahuddin oil refinery in Iraq. The soft sensor was designed using several stages, including data collection, preprocessing, clustering, feature selection, and classification. The proposed TWOA achieved a higher fault classification result of 99.98% compared to other algorithms.

Keywords-soft sensor; optimization; WOA; tent map; TWOA

I. INTRODUCTION

The petrochemical sector faces significant challenges from market demands and environmental authorities that require enhanced product quality and energy efficiency. These challenges are the driving forces for a new round of scientific and technological developments in production and operations management in the petrochemical sector, including the exploitation of smart IT networks, sensors, online process analytical technology, and advanced data analytics [1-3]. This paper presents a new development in advanced data analytics that adds new capabilities to refineries and supports their continuous efforts to meet the aforementioned challenges for operational excellence.

An efficient way to address the issue of important variables that are difficult to measure in intricate industrial processes is through the use of soft sensor technology. Data cannot be adequately represented by supervised or unsupervised modeling techniques alone, resulting in less-than-ideal outcomes when applying learned soft sensor models to the prediction of crucial variables [4]. In contemporary industrial processes, data-driven methods, particularly those based on deep learning, have attracted a lot of interest in quality prediction. However, in reality, industrial data are frequently produced in data streams. The dynamic nature of industrial data leads to a deteriorated model prediction ability when operating circumstances and process parameters vary with the environment. However, a deep transfer learning-based approach can automatically extract domain-invariant features in a variety of tasks to adapt to changes in data distribution [5].

Numerous petroleum subproducts are produced in refineries, which have a range of uses and must fulfill a set of safety and quality requirements [6]. Flashpoint Temperature (FT) is one of the most important quality requirements for automobile diesel [7]. In general, there are two types of soft sensor models: data-driven models that are based on historical data and first-principle models that are based on process mechanisms [8]. Data-driven soft sensor models have demonstrated clear benefits in intricate industrial processes, as Distributed Control System (DCS) applications simplified data storage and simplified the process mechanism [9]. Consequently, a greater variety of widely used techniques, including Support Vector Machines (SVM) [10], Partial Least Squares Regression (PLSR) [11], and Principal Component Regression (PCR) [12], have been widely used in data-driven soft sensor modeling. However, a global optimization technique is essential for further improvement of feature selection.

Table I presents recent studies on artificial intelligence, machine learning, deep learning, feature selection, and soft sensors, along with the techniques used. This study addresses the issues of previous ones by proposing an efficient feature selection algorithm based on the Whale Optimization Algorithm (WOA) and tent chaotic map to select the best relevant features to increase the accuracy of a fault detection system.

TABLE I. COMPARISON OF VARIOUS RELATED WORKS

Ref.	Year	Methods used
[13]	2023	BiLSTM network with attention mechanism for spatiotemporal feature extraction.
[14]	2022	Feature selection using PCC and VIF; model generation with SVM and RF.
[15]	2019	Stacked auto-encoders for variable selection in deep learning models.
[16]	2021	Deep kernel extreme learning machine for feature extraction and modeling.
[17]	2023	Recursive Feature Elimination (RFE) for feature selection and machine learning models.
[18]	2022	Semi-supervised autoencoders for active learning and model updating.
[19]	2023	Multi-Layer Perceptron (MLP) for embedded feature and sensor selection.
[20]	2023	Causal model-based feature selection for industrial processes.
[21]	2020	Survey on filter, wrapper, and embedded methods for input selection in soft sensor development.
[22]	2015	Denoising Auto-Encoders (DAE) for feature extraction + BackPropagation Neural Networks (BPNN) for regression.
[23]	2022	Machine learning models for estimating chlorophyll-a concentrations using low-cost input variables.
[24]	2023	AI-driven algorithms - deep learning and fuzzy logic for process monitoring and control.
[25]	2020	Echo State Network (ESN) optimized by an improved genetic algorithm for feature selection combined with SVR for predictive modeling.

II. THE PROPOSED APPROACH

The proposed approach was designed to detect all fault types, which is the main challenge that was not achieved in previous works. This method includes several stages that work together to build the desired soft-sensor model, such as data collection, preprocessing, clustering, feature selection, and classification. Figure 1 shows the main stages of the proposed method.

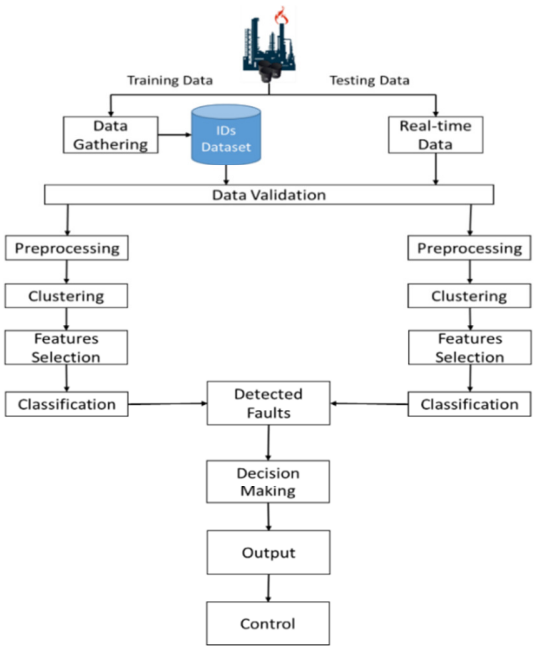

Fig. 1. The proposed method.

Most of the stages are described in [26]. This paper presents the proposed feature selection approach in detail. One of the most important phases of this research was gathering data regarding the operational and control systems in the oil refinery. Real data were collected from the Salahuddin oil refinery in Iraq, as shown in Figure 2.

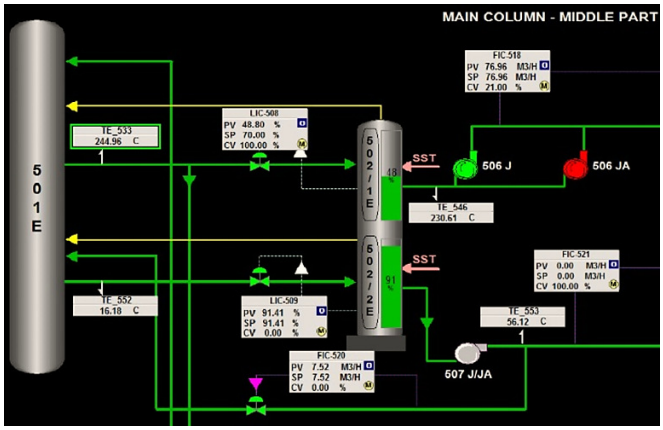

Fig. 2. Salahuddin oil refinery.

Data were collected for seven months, from Jan. 1 2023 to Jul. 31, 2023. Every ten seconds, data were collected from the daily report of the unit's activities. Subsequently, the data were validated and exported from an Excel spreadsheet to a CSV file. The datasheet contains several types of data: Temperature (TE), Pressure (PV), data flow, Set Point (SP), Control Valve (CV), and Actual Value (AV). The most effective values considered were TE and PV, which are correlated with the SP using Pearson Correlation Coefficient Analysis (PCCA) to meet the desired decisions for the CV [27]. The total number of collected data was 20 million. Table II presents a sample of the collected data.

TABLE II. SAMPLE OF THE COLLECTED DATA FROM SALAHUDDIN OIL REFINERY

TE	PV	SP	CV
47.88	67.05	66	15.15
202.09	73.97	67	100
205.49	84.82	84.82	0
20.26	8.29	8.29	100
240.49	259.35	260	63.45
255.47	199.95	200	62.66
20.84	0	0	100

A. Feature Selection

The proposed system uses an enhanced version of the WOA, called TWOA, based on a tent chaotic map, to select the most relevant features in the dataset.

1) Whale Optimization Algorithm (WOA)

WOA is a nature-inspired metaheuristic optimization algorithm that mimics the social behavior of humpback whales. WOA is one of the most popular population-based metaheuristic algorithms for solving global optimization problems in a variety of domains [28].

The humpback whale's natural hunting behavior serves as the model for this algorithm. At the water's surface, humpback whales hunt by focusing on schools of krill or tiny fish. They encircle and ensnare their victim by forming characteristic bubbles along a spiral. WOA simulates the behavior of whales utilizing three strategies: encircling the prey, searching for prey (exploration phase), and spiral bubble-net attacking (exploitation phase). The position of the i-th whale at iteration t is denoted by $X_i^t = (x_{i,1}^t, x_{i,2}^t, \dots, x_{i,D}^t)$, where $i = (1, 2, \dots, N)$ and N and D are the whale population and the dimensions of the problem, respectively. The following subsections present mathematically the strategy of WOA.

a) Encircling Prey

The ability to locate prey allows humpback whales to surround them. WOA considers that the most suitable option is that the target prey is near the ideal solution, as it is impossible to know in advance where the ideal design is located in the search space. The other search agents will thus attempt to adjust their locations to align with the agent with the greatest efficiency once it has been determined. The following equations illustrate this behavior:

$$\vec{D} = \left|\vec{C}.\vec{X}^*(t) - \vec{X}(t)\right| \quad (1)$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A}.\vec{D} \quad (2)$$

where $|\ |$ is the value in absolute terms, t denotes the current iteration, A and C are coefficient vectors, and X^* is the position of the most effective solution thus far. It is important to note that if a better solution is found, X^* should be modified in every iteration. The vectors $\vec{A}$ and $\vec{C}$ are calculated as follows:

$$\vec{A} = 2\vec{a}.\vec{r} - \vec{a} \quad (3)$$

$$\vec{C} = 2.\vec{r} \quad (4)$$

where $\vec{r}$ is a random vector in [0, 1] and $\vec{a}$ is linearly reduced from 2 to 0 during iterations (in both exploration and exploitation phases). Equation (2) mimics surrounding the prey and enables any search agent to update its location in the vicinity of the current best answer. The same idea can be used in a search space with n dimensions, where the search agents will circle the best answer thus far in hyper-cubes.

b) Bubble-net Attacking Method (Exploitation Phase)

To mathematically model the bubble-net behavior of humpback whales, two approaches are used:

- Shrinking encircling mechanism: The value of $\vec{a}$ in (3) is decreased to obtain this behavior. Keep in mind that $\vec{a}$ reduces the fluctuation range of $\vec{A}$. In another way, $\vec{A}$ is a random number in the range $[-a, a]$, where a is reduced throughout the series of repetitions from 2 to 0. The updated position of the searching agent can be specified everywhere between the agent's initial position and the position of the most effective agent at that moment, provided that $\vec{A}$ is set to random values in $[-1, 1]$. Figure 3 displays various locations from (X, Y) into (X^*, Y^*) that can be attained by $0 \leq A \leq 1$ in a 2D space.

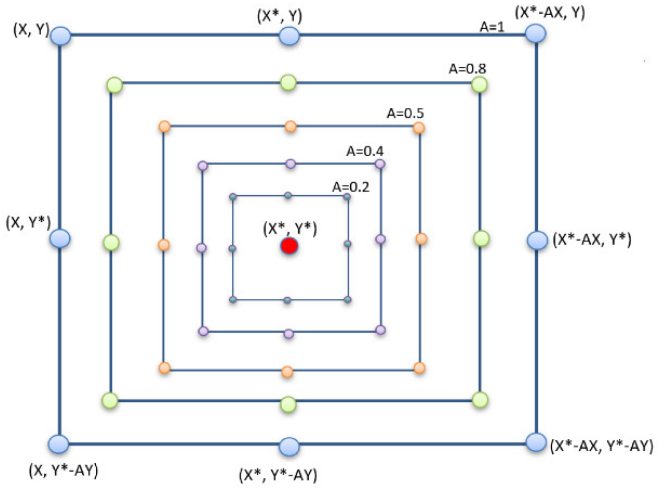

Fig. 3. The encircling mechanism.

- Spiral updating position: The gap between the whale at (X, Y) and the prey at (X^*, Y^*) is initially determined using this method. To replicate the helix-shaped motion observed by humpback whales, a spiral equation connecting the whale's and prey's positions is then constructed as follows:

$$\vec{X}(t+1) = \overrightarrow{D'}.e^{bl}.\cos(2\pi l) + \overrightarrow{X^*}(t) \quad (5)$$

where $\overrightarrow{D'} = |\overrightarrow{X^*}(t) - \vec{X}(t)|$ and denotes the distance from the i-th whale to the prey (best solution obtained so far), b is a constant for defining the shape of the logarithmic spiral, and l is a random number in $[-1,1]$. Figure 4 shows the spiral approach.

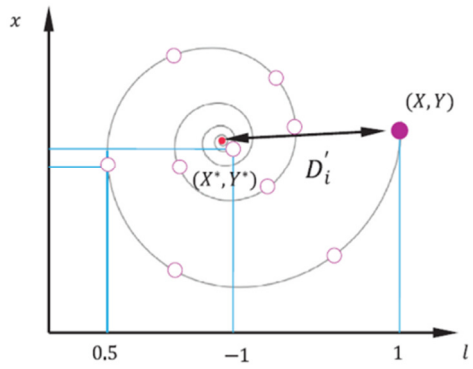

Fig. 4. The spiral updated location.

It should be noted that humpback whales move in a spiral pattern and a decreasing radius around their prey. We suppose that there's a 50% chance of selecting either the spiral model or the narrowing enclosing mechanism for adjusting the whales' position throughout optimization to mimic this continuous behavior. The following is the mathematical model:

$$\vec{X}(t+1) = \begin{cases} \overrightarrow{X^*}(t) - \vec{A}.\vec{D} & \text{if } p < 0.5 \\ \overrightarrow{D'}.e^{bl}.\cos(2\pi l) + \overrightarrow{X^*}(t) & \text{if } p \geq 0.5 \end{cases} \quad (6)$$

where p in $[0,1]$ is a randomized value. Along with using a bubble net, humpback whales also conduct haphazard prey searches.

c) Search for Prey (Exploration Phase)

To find prey (searching), the same strategy using the modification of the $\vec{A}$ vector can be applied. In reality, humpback whales search at random based on one another's positions. To push the search agent to wander far away from a reference whale, $\vec{A}$ is employed with values that are random larger than 1 or less than -1. In the exploration stage, as opposed to the exploitation phase, a search agent's location is modified based on a randomly selected search agent rather than the best search agent thus far. The WOA algorithm can do a global search thanks to this technique and $|\vec{A}| > 1$. The following is the mathematical representation:

$$\vec{D} = |\vec{C}.\overrightarrow{X_{rand}} - \vec{X}| \quad (7)$$

$$\vec{X}(t+1) = \overrightarrow{X_{rand}} - \vec{A}.\vec{D} \quad (8)$$

where $\overrightarrow{X_{rand}}$ is a random position vector (a random whale) chosen from the current population. A collection of random solutions is used to launch the WOA algorithm. Search agents adjust their locations to either a randomly selected search agent or the most effective solution found thus far at every repetition. Exploration and extraction are provided by decreasing a value from 2 to 0. To update the search agents' positions, the best solution is picked when $|\vec{A}| < 1$ and a random search agent is chosen when $|\vec{A}| > 1$. WOA can alternate between a spiral and a circular movement, determined by the value of p. When a termination requirement is satisfied, the WOA algorithm is finally ended.

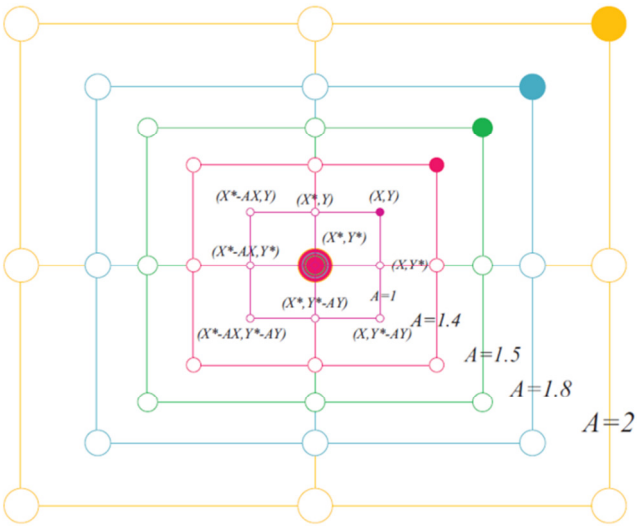

Fig. 5. Exploration mechanism.

Algorithm 1 describes the WOA algorithm. Theoretically, as WOA incorporates exploration and extraction capabilities, it may be regarded as a global optimizer. Additional search agents can use the most recent optimal record inside the defined search space in the vicinity of the best answer

according to the suggested hypercube method. WOA may seamlessly switch between exploration and extraction due to responsive variation of the searching vector A. By reducing A, certain repetitions are allocated to exploration ($|\vec{A}| \geq 1$) and the remaining to exploitation ($|\vec{A}| < 1|$). Interestingly, WOA has only two primary inner parameters (A and C) that need to be changed.

```
Algorithm 1 WOA
Input: Populations
Output: Best solution
Initialize the whales population Xi (i =
1, 2, ..., n)
Calculate the fitness of each search agent
X*: the best search agent
While (t < maxno of iterations)
  For each search agent
    Update a, A, C, l, and p
    If (p<0.5)
      If (|A|< 1)
        Update position using (1)
      Else
        Select a random search agent
      Update position using (8)
      Endif
    Else
      Update position using (5)
    Endif
  Endfor
  Check if any search agent goes beyond
  the Search space and amend it
  Calculate the fitness of each search
  agent
  Update X* if there is a better solution
  t=t+1
Endwhile
Return X*
```

2) Tent Chaotic Map

The mathematical procedures known as chaotic maps use a starting seed value to produce a very random pattern. In the mathematical analysis of chaotic maps, dynamic systems generate an arbitrary state that is completely disorganized, looks irregular, and is controlled by the original seed requirements. The link between entirely unpredictable chaotic results and the fundamental patterns that produce them is explained by chaos theory. Knowing how a generator is connected allows for a thorough analysis of these patterns. These generators usually depend on self-similarity, consistency, feedback loops, and the system's fractal characteristics. Figure 6 presents some chaotic maps. WOA suffers from premature convergence, which makes it stuck in local optima, low diversity when initializing the population, and slow convergence speed in a complex research space. WOA randomness and exploration capabilities could be enhanced using a chaotic map. The proposed approach uses a tent map to enhance the randomness and exploration capabilities of WOA. A tent map illustrates the extremely precise consequences of chaotic behavior. The following equation represents the chaotic sequence phrases in this function:

$$x_{k+1} = \begin{cases} 2x_{k'} & x_k < 0.5, \\ 2(1 - x_k), & x_k \geq 0.5, \end{cases} \quad (9)$$

where x_k ranges from 0 to 1. The tent map generates chaotic sequences in (0, 1).

The tent map has a light computation explicit formula that makes it fast and simple. In addition, the tent map offers a more uniform coverage in the search space of WOA via the evenly distributed chaotic sequence. Also, it helps WOA avoid local optima by initializing a diverse population within a few control parameters compared to other chaotic maps. These enhancements directly affect the performance of the fault detection system by making it robust and accurate by creating a clear and functional correlation with the system.

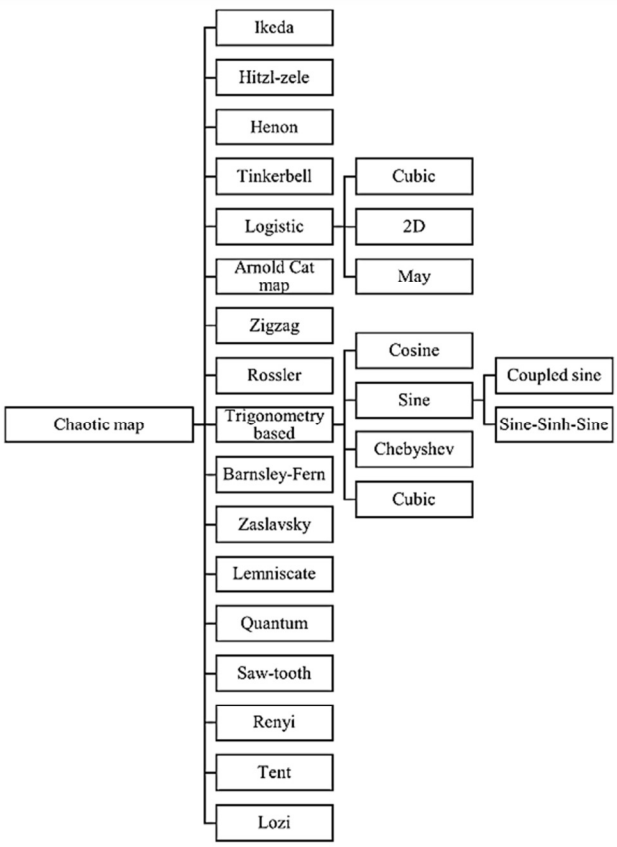

Fig. 6. Different chaotic maps.

3) Proposed Feature Selection Approach (TWOA)

TWOA has two main differences from WOA: (i) Initializes the population by producing values in control parameters using the tent chaotic map, and (ii) Updates whale position using modified equations. Figure 7 illustrates the main steps of the proposed feature extraction approach. The tent map is used to initialize the population with the best possible diversity. This initialization leads to better results in selecting the best whale position and avoiding the local optima. Figure 8 illustrates the Tent map. Update position (2) is modified as follows:

$$\vec{X}(t+1) = \frac{|\vec{X}*(t) - \vec{A}\,.\,\vec{D})|}{2} \quad (10)$$

Dividing (2) by two enhances the performance of updating the position. This approach reduces the step size into the prey and improves the exploitation through accurate local search at the last iterations of the algorithm. This also improves the stability in the late-stage convergence. Position update (8) is modified as follows:

$$\vec{X}(t+1) = \frac{|\overrightarrow{X_{Rand}}*(t) - \vec{A}\,.\,\vec{D})|}{2*A*R} \quad (11)$$

where R is a random number between $(0,1)$ to enhance the performance of the modified equation.

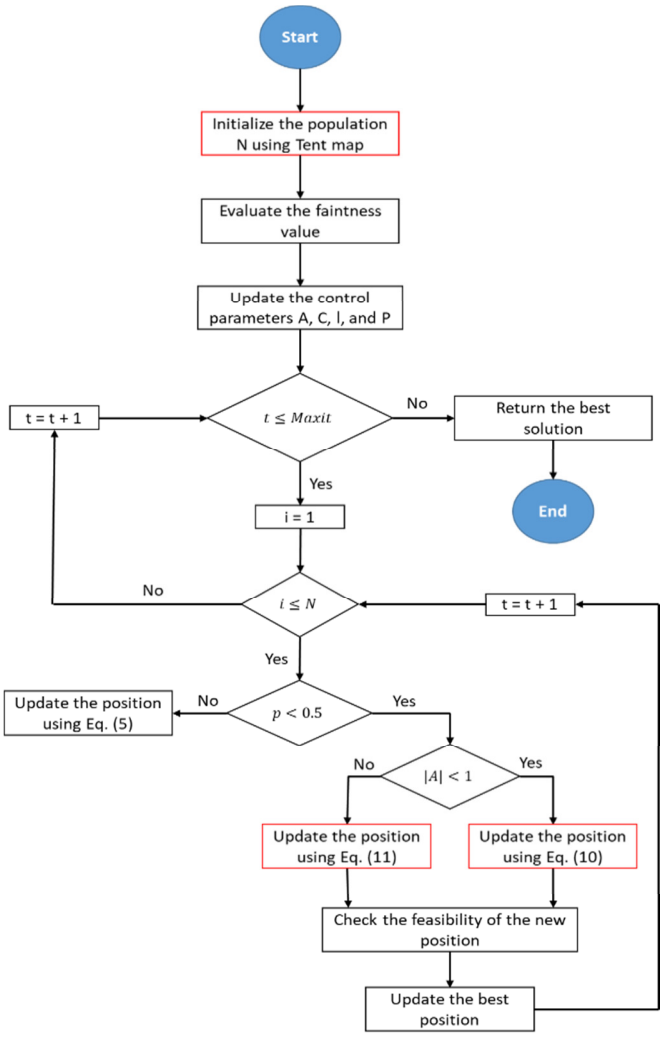

Fig. 7. Proposed TWOA.

Dividing by $(2 * A * R)$ provides an adaptive step size control since R and A change over time. R adds a stochastic scale and enhances the diversity of the search path. This offers further randomness, making the Whale spiral different every time. Equation (11) provides a better exploration to avoid local minima and premature convergence due to the slight random movement each time. Since R changes every time, it offers a dynamic balance between exploration and exploitation. The diversity offered by the proposed formula makes the whales move in different ways (even close whales), provides necessary randomness into the whales' spiral motion, and enhances the exploration capabilities of WOA. This approach prevents premature convergence, maintains search diversity, and enables better global optimization performance, particularly for multi-modal, complex optimization landscapes.

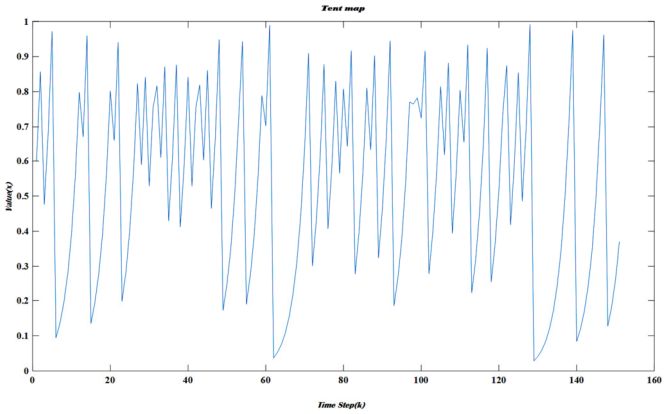

Fig. 8. Tent map.

```
Algorithm 2: Proposed TWOA
Input: Populations
Output: Best solution
Initialize the whale population
Xi (i=1....n) using Tent map
Determine the fitness function f(x). X* is
the best search agent
While (t < MaxItr)
  For each search agent
    Update A, C, a, l, and P
    If (p<0.5)
      If (|A|<1)
        Update the position using (10)
      Else
        Select random agent using (11)
        Update the position
      Endif
    Else
      Update position using (5)
    EndiF
  Endfor
  Check if any search agent goes beyond
  the Search space and amend it
  Calculate the fitness
  Update X*
  t=t+1
Endwhile
Return X*
```

B. Performance Evaluation

Different evaluation metrics can be used to evaluate the performance of the proposed approach. This study used the accuracy metric, which is given by:

$$Accuracy \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (12)$$

III. RESULTS AND DISCUSSION

The proposed approach was implemented in MATLAB 2023b in Windows 11. Several methods have been used to evaluate the performance of soft sensor systems. The input dataset from the CSV file was divided into two sets: 70% for training and 30% for testing. The proposed approach was used with different classifiers, and Table III shows their accuracy.

TABLE III. RESULTS WITH DIFFERENT CLASSIFIERS

Classifier	Accuracy
KNN	97.5%
ANN	97.2%
Tree	96.5%
SVM	99.2%
BDT-SVM	99.98%

TABLE IV. COMPARISON RESULTS OF TWOA AND OTHER OPTIMIZATION ALGORITHMS

Algorithm	Accuracy
DE	92%
PSO	94.7%
GSA	96.3%
GWO	97.5%
WOA	97.2%
TWOA	99.98%

The proposed approach obtained a 99.98% accuracy using BDT-SVM, outperforming the other machine learning classifiers. In addition, the proposed approach achieved the highest accuracy compared to other optimization algorithms, as shown in Table IV. The proposed TWOA outperformed other well-known optimization algorithms, namely DE by 7.98%, PSO by 5.28%, and GSA by 3.68%. WOA and GWO achieved high accuracies of 97.2% and 97.5%, respectively. However, the proposed TWOA outperformed both of them by 2.48% and 2.78%, respectively.

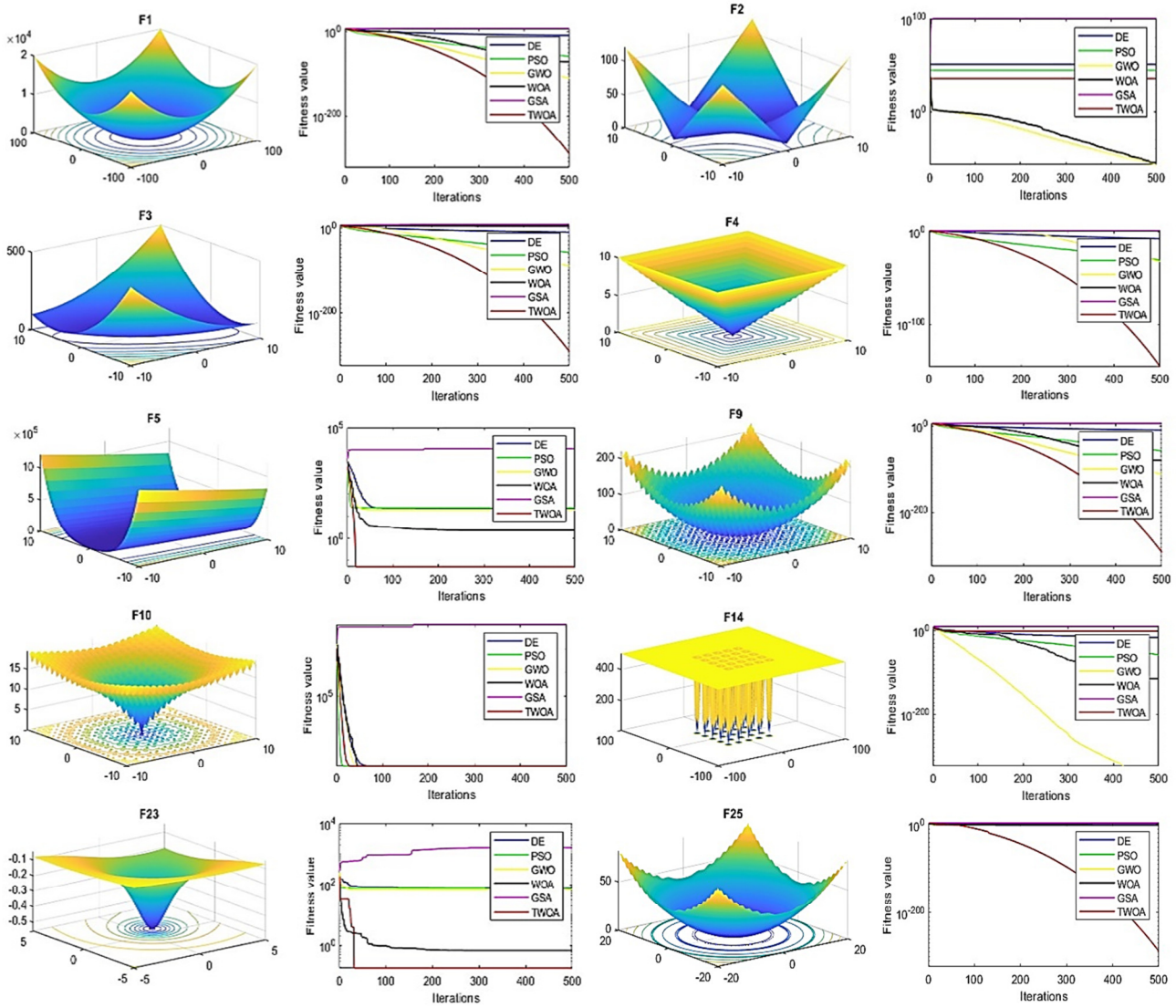

Fig. 9. Results using different benchmark functions.

To further evaluate the performance of TWOA, 28 benchmark functions were used. After many iterations, the results show that the proposed approach achieved the best solutions in 24 benchmark functions, as shown in Figure 9. TWOA achieved the best solutions in (F1, F3-F9, F11-F13, F15-F21, and F23-F28). However, in (F2, F10, F14, and F22) GWO and WOA achieved the best solutions.

Another experiment was carried out to evaluate the performance of TWOA using five standard datasets from the UCI Machine Learning Repository [28] (Sonar, Waveform, Spect, Ionosphere, and Spambase) and the SVM classifier [29]. Dataset details are listed in Table V.

TABLE V. DATASET DETAIL

No.	Dataset	No. of features	No. of instances
1	Sonar	60	208
2	Waveform	21	5000
3	Spect	22	267
4	Ionosphere	34	351
5	Spambase	57	4601

TWOA was tested in all five datasets and achieved higher classification accuracy than the original WOA in all cases. Table VI shows the accuracy results of applying TWOA and WOA on the UCI datasets.

TABLE VI. RESULTS ON UCI DATASETS

No.	Dataset	TWOA%	WOA%
1	Sonar	0.9816	0.9022
2	Waveform	0.8721	0.8707
3	Spect	0.8870	0.8757
4	Ionosphere	0.9577	0.9234
5	Spambase	0.9478	0.9161

A limitation of the proposed approach is the computation time of (10). Dividing the original update position equation leads to stable and accurate results at the cost of a little increase in the computation time. Table VII shows the computation time of the original and proposed equations. However, WOA requires 600 iterations to converge, while TWOA needs 800 iterations to converge.

TABLE VII. THE EXPERIMENTAL OF THE DATASETS

Algorithm	Time per iteration	Total computation time
WOA	5 ms	3 s (for 600 iterations)
TWOA	5.1 ms	4 s (for 800 iterations)

IV. CONCLUSION

This paper presented an improved version of WOA, called TWOA. This algorithm enhances the performance of the fault detection and classification system to detect and classify all fault types that occur in an oil refinery. The WOA was improved using a two-stage approach: First, by modifying the updating position equation and then using a tent map to initialize a diversified population for the next steps. TWOA achieved a higher classification accuracy of 99.98% compared to other algorithms. Using BDT-SVM for classification achieved higher accuracy by 0.78% compared to other classifiers. TWOA was compared to other popular optimization algorithms, such as DE, PSO, GSA, and GWO, achieving a 2-7% better accuracy. The TWOA results on 24 of 28 benchmarks were the best. In addition, using TWOA in feature selection achieved the highest accuracy on five standard datasets from UCI.

ACKNOWLEDGMENT

The authors would like to thank Universiti Putra Malaysia (UPM), the Malaysian Ministry of Higher Education (MOHE), Al-Iraqia University, and the Iraqi Ministry of Higher Education and Scientific Research for their continuous support in the research work. This work was supported in part by North Refineries Company and ALNUHA for General Contracts & Industrial Equipments Trading Ltd.

REFERENCES

[1] T. Lemos *et al.*, "Echo State Network Based Soft Sensor for Monitoring and Fault Detection of Industrial Processes," *Computers & Chemical Engineering*, vol. 155, Dec. 2021, Art. no. 107512, https://doi.org/10.1016/j.compchemeng.2021.107512.

[2] D. C. M. de Souza, L. Cabrita, C. F. Galinha, T. J. Rato, and M. S. Reis, "A Spectral AutoML approach for industrial soft sensor development: Validation in an oil refinery plant," *Computers & Chemical Engineering*, vol. 150, Jul. 2021, Art. no. 107324, https://doi.org/10.1016/j.compchemeng.2021.107324.

[3] M. S. Reis and G. Gins, "Industrial Process Monitoring in the Big Data/Industry 4.0 Era: from Detection, to Diagnosis, to Prognosis," *Processes*, vol. 5, no. 3, Sep. 2017, Art. no. 35, https://doi.org/10.3390/pr5030035.

[4] S. Gao, J. Xu, Z. Ma, R. Tian, X. Dang, and X. Dong, "Research on Modeling of Industrial Soft Sensor Based on Ensemble Learning," *IEEE Sensors Journal*, vol. 24, no. 9, pp. 14380–14391, Feb. 2024, https://doi.org/10.1109/JSEN.2024.3375072.

[5] Y. Wang, H. Jin, B. Wang, and B. Yang, "A Deep Learning Soft Sensor Based on Domain-Invariant Features Extraction and Online Local Adaptation," in *2024 IEEE 13th Data Driven Control and Learning Systems Conference (DDCLS)*, Kaifeng, China, May 2024, pp. 1311–1317, https://doi.org/10.1109/DDCLS61622.2024.10606794.

[6] B. Karun, R. V. R., and S. Elayidom, "Application of fuzzy logic and machine learning techniques to improve inherently safer design in process safety management: A brief study," *Process Safety Progress*, vol. 41, no. S1, pp. S178–S186, 2022, https://doi.org/10.1002/prs.12331.

[7] H. E. Ahmed, I. F. Tahoun, and S. Zakel, "Development of decahydronaphthalene reference material for low flash point measurements," *Egyptian Journal of Petroleum*, vol. 30, no. 1, pp. 7–10, Mar. 2021, https://doi.org/10.1016/j.ejpe.2020.12.002.

[8] Z. Li, H. Jin, S. Dong, B. Qian, B. Yang, and X. Chen, "Semi-supervised ensemble support vector regression based soft sensor for key quality variable estimation of nonlinear industrial processes with limited labeled data," *Chemical Engineering Research and Design*, vol. 179, pp. 510–526, Mar. 2022, https://doi.org/10.1016/j.cherd.2022.01.026.

[9] W. Shao, C. Xiao, J. Wang, D. Zhao, and Z. Song, "Real-time estimation of quality-related variable for dynamic and non-Gaussian process based on semisupervised Bayesian HMM," *Journal of Process Control*, vol. 111, pp. 59–74, Mar. 2022, https://doi.org/10.1016/j.jprocont.2022.01.007.

[10] M. Lee, J. Bae, and S. B. Kim, "Uncertainty-aware soft sensor using Bayesian recurrent neural networks," *Advanced Engineering Informatics*, vol. 50, Oct. 2021, Art. no. 101434, https://doi.org/10.1016/j.aei.2021.101434.

[11] Q. Sun and Z. Ge, "A Survey on Deep Learning for Data-Driven Soft Sensors," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 5853–5866, Sep. 2021, https://doi.org/10.1109/TII.2021.3053128.

[12] M. Maggipinto, A. Beghi, and G. A. Susto, "A Deep Convolutional Autoencoder-Based Approach for Anomaly Detection With Industrial,

Non-Images, 2-Dimensional Data: A Semiconductor Manufacturing Case Study," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1477–1490, Jul. 2022, https://doi.org/10.1109/TASE.2022.3141186.

[13] C. Xie, R. Yao, L. Zhu, H. Gong, H. Li, and X. Chen, "Soft-Sensor Development through Deep Learning with Spatial and Temporal Feature Extraction of Complex Processes," *Industrial & Engineering Chemistry Research*, vol. 62, no. 1, pp. 519–534, Jan. 2023, https://doi.org/10.1021/acs.iecr.2c03137.

[14] W. Y. Moon and S. D. Kim, "A Framework of Soft Sensor Systems with Machine Learning," in *2022 International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA, Dec. 2022, pp. 153–159, https://doi.org/10.1109/CSCI58124.2022.00031.

[15] X. Wang, "A New Variable Selection Method for Soft Sensor Based on Deep Learning," in *2018 5th IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, Nanjing, China, Nov. 2018, pp. 674–678, https://doi.org/10.1109/CCIS.2018.8691405.

[16] Y. Meng, J. Chen, Z. Li, Y. Zhang, L. Liang, and J. Zhu, "Soft sensor with deep feature extraction for a sugarcane milling system," *Journal of Food Process Engineering*, vol. 45, no. 8, 2022, Art. no. e14066, https://doi.org/10.1111/jfpe.14066.

[17] H. Y. Shyu, C. J. Castro, R. A. Bair, Q. Lu, and D. H. Yeh, "Development of a Soft Sensor Using Machine Learning Algorithms for Predicting the Water Quality of an Onsite Wastewater Treatment System," *ACS Environmental Au*, vol. 3, no. 5, pp. 308–318, Sep. 2023, https://doi.org/10.1021/acsenvironau.2c00072.

[18] D. Cacciarelli, M. Kulahci, and J. Tyssedal, "Online Active Learning for Soft Sensor Development using Semi-Supervised Autoencoders." arXiv, Apr. 09, 2023, https://doi.org/10.48550/arXiv.2212.13067.

[19] A. Saha and N. R. Pal, "Group-feature (Sensor) selection with controlled redundancy using neural networks," *Neurocomputing*, vol. 610, Dec. 2024, Art. no. 128596, https://doi.org/10.1016/j.neucom.2024.128596.

[20] Y. N. Sun, W. Qin, J. H. Hu, H. W. Xu, and P. Z. H. Sun, "A Causal Model-Inspired Automatic Feature-Selection Method for Developing Data-Driven Soft Sensors in Complex Industrial Processes," *Engineering*, vol. 22, pp. 82–93, Mar. 2023, https://doi.org/10.1016/j.eng.2022.06.019.

[21] F. Curreri, G. Fiumara, and M. G. Xibilia, "Input Selection Methods for Soft Sensor Design: A Survey," *Future Internet*, vol. 12, no. 6, Jun. 2020, Art. no. 97, https://doi.org/10.3390/fi12060097.

[22] Yujun Lin and W. Yan, "Study of soft sensor modeling based on deep learning," in *2015 American Control Conference (ACC)*, Chicago, IL, USA, Jul. 2015, pp. 5830–5835, https://doi.org/10.1109/ACC.2015.7172253.

[23] A. Mozo *et al.*, "Chlorophyll soft-sensor based on machine learning models for algal bloom predictions," *Scientific Reports*, vol. 12, no. 1, Aug. 2022, Art. no. 13529, https://doi.org/10.1038/s41598-022-17299-5.

[24] Y. S. Perera, D. A. A. C. Ratnaweera, C. H. Dasanayaka, and C. Abeykoon, "The role of artificial intelligence-driven soft sensors in advanced sustainable process industries: A critical review," *Engineering Applications of Artificial Intelligence*, vol. 121, May 2023, Art. no. 105988, https://doi.org/10.1016/j.engappai.2023.105988.

[25] R. Huang, Z. Li, and B. Cao, "A Soft Sensor Approach Based on an Echo State Network Optimized by Improved Genetic Algorithm," *Sensors*, vol. 20, no. 17, Jan. 2020, Art. no. 5000, https://doi.org/10.3390/s20175000.

[26] M. F. S. AlRijeb, M. L. Othman, A. Ishak, M. K. Hassan, and B. M. Albaker, "Machine Learning-Driven Soft Sensor Implementation for Real-Time Fault Detection in CDU of Oil Refinery," *Engineering, Technology & Applied Science Research*, vol. 15, no. 1, pp. 20425–20432, Feb. 2025, https://doi.org/10.48084/etasr.9781.

[27] R. F. Tate, "Correlation Between a Discrete and a Continuous Variable. Point-Biserial Correlation," *The Annals of Mathematical Statistics*, vol. 25, no. 3, pp. 603–607, 1954.

[28] A. Thiruneelakandan, G. Kaur, G. Vadnala, N. Bharathiraja, K. Pradeepa, and M. Retnadhas, "Measurement of oxygen content in water with purity through soft sensor model," *Measurement: Sensors*, vol. 24, Dec. 2022, Art. no. 100589, https://doi.org/10.1016/j.measen.2022.100589.

[29] P. Bangert, "Soft sensors for NOx emissions," in *Machine Learning and Data Science in the Power Generation Industry*, P. Bangert, Ed. Elsevier, 2021, pp. 213–226.