



Article Feature Selection and Classification of Transformer Faults Basedon Novel Meta-Heuristic Algorithm

El-Sayed M. El-kenawy ¹, Fahad Albalawi ², Sayed A. Ward ^{3,4}, Sherif S. M. Ghoneim ², Marwa M. Eid ⁵, Abdelaziz A. Abdelhamid ^{6,7}, Nadjem Bailek ⁸ and Abdelhameed Ibrahim ^{9,*}

- ¹ Department of Communications and Electronics, Delta Higher Institute of Engineering and Technology, Mansoura 35111, Egypt
- ² Electrical Engineering Department, College of Engineering, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia
- ³ Electrical Engineering Department, Shoubra Faculty of Engineering, Benha University, 108 Shoubra St., Cairo 11629, Egypt
- ⁴ Faculty of Engineering, Delta University for Science and Technology, Mansoura 11152, Egypt
- ⁵ Faculty of Artificial Intelligence, Delta University for Science and Technology, Mansoura 11152, Egypt ⁶ Department of Computer Science, College of Computing and Information Technology, Spagra University
- ⁶ Department of Computer Science, College of Computing and Information Technology, Shaqra University, Shaqra 11961, Saudi Arabia
- ⁷ Department of Computer Science, Faculty of Computer and Information Sciences, Ain Shams University, Cairo 11566, Egypt
- ⁸ Energies and Materials Research Laboratory, Faculty of Sciences and Technology, University of Tamanghasset, Tamanrasset 11001, Algeria
- ⁹ Computer Engineering and Control Systems Department, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt
- Correspondence: afai79@mans.edu.eg

Abstract: Detecting transformer faults is critical to avoid the undesirable loss of transformers from service and ensure utility service continuity. Transformer faults diagnosis can be determined based on dissolved gas analysis (DGA). The DGA traditional techniques, such as Duval triangle, Key gas, Rogers' ratio, Dornenburg, and IEC code 60599, suffer from poor transformer faults diagnosis. Therefore, recent research has been developed to diagnose transformer fault and the diagnostic accuracy using combined traditional methods of DGA with artificial intelligence and optimization methods. This paper used a novel meta-heuristic technique, based on Gravitational Search and Dipper Throated Optimization Algorithms (GSDTO), to enhance the transformer faults' diagnostic accuracy, which was considered a novelty in this work to reduce the misinterpretation of the transformer faults. The robustness of the constructed GSDTO-based model was addressed by the statistical study using Wilcoxon's rank-sum and ANOVA tests. The results revealed that the constructed model enhanced the diagnostic accuracy up to 98.26% for all test cases.

Keywords: diagnostic accuracy; transformer faults; Gravitational Search; artificial intelligence

MSC: 68T07; 68T10; 68T20

1. Introduction

The power transformer considers one of the most vital elements in the electrical power system, as the wrong or repeated disconnection leads to the loss of a lot of profits for the electricity companies [1,2]. The undesirable outage of the transformer from the electrical power system occurs as a result of the exposure of the insulation system, which is the insulation oil and insulation paper, to various stresses, whether electrical, thermal, or mechanical. These stresses lead to insulation damage and rapid deterioration, which requires early prediction of the deterioration of the insulation condition inside the transformer. Dissolved gas analysis (DGA) is a common technique that was used for detecting and exploring the transformer faults based on the dissolved gases concentrations [3–5].



Citation: El-kenawy, E.-S.M.; Albalawi, F.; Ward, S.A.; Ghoneim, S.S.M.; Eid, M.M.; Abdelhamid, A.A.; Bailek. N.; Ibrahim, A. Feature Selection and Classification of Transformer Faults Based on Novel Meta-Heuristic Algorithm. *Mathematics* **2022**, *10*, 3144. https://doi.org/10.3390/ math10173144

Academic Editors: Rui Peng, Kaiye Gao, Ioannis G. Tsoulos and Nicu Bizon

Received: 29 June 2022 Accepted: 29 August 2022 Published: 1 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). Several traditional DGA techniques, such as Key gases, Dornenburg, Rogers' ratios, IEC code, and Duval triangle, contribute to interpreting the cause of transformer faults [6–10]. The DGA methods above were developed to diagnose the transformer faults based on the combustible gases' ratios [4]. The combustible gases are Hydrogen (H_2), Methane (CH_4), Ethan (C_2H_6), Ethylene (C_2H_4), and Acetylene (C_2H_2). Some of these methods used four ratios between the five gases, such as Dornenburg and Rogers' four ratios and the other methods used three ratios, such as IEC60599 code [11–13]. The Duval triangle method is one of the most common DGA techniques which has been used until now for diagnosing transformer faults, which is based the diagnose technique on the three gases (Methane (CH_4), Ethylene (C_2H_4), and Acetylene (C_2H_2)) [8,9]. The limitation of the traditional DGA techniques is the failure to interpret the transformer faults when the cases are out of the code. New graphical DGA techniques were developed to enhance the diagnostic accuracy of the transformer faults, such as Pentagon [14–16] and heptagon [3].

Researchers have recently merged artificial intelligence with the traditional DGA techniques to enhance its diagnostic accuracy. Artificial Neural Networks (ANNs) were utilized to build a smart system to enhance the diagnostic accuracy by comparing the output of five traditional DGA techniques to identify the dominant transformer faults. In addition, the ANN was used in research for the same purpose [17–21]. Fuzzy logic was also merged with traditional DGA techniques to enhance diagnostic accuracy, and several publications were reported [22,23]. A neuro-fuzzy was developed in [24] to enhance the transformer fault diagnostic accuracy. Support vector machine (SVM) and other classifiers were also used based on the feature selection for each fault to diagnose the transformer faults [25,26].

Several kinds of research addressed the utilization of optimization techniques with DGA to adapt the ratio limits of the traditional IEC Code and Rogers' ratio methods to enhance the diagnostic accuracy of the transformer faults. The Particle Swarm Optimization (PSO) algorithm in [27] was used to modify the limits of the ratios among the gases to overcome the diagnostic failure of the transformer faults. It is also addressed in several works, as in [28,29] that used PSO with the DGA techniques to enhance the transformer faults' diagnostic accuracy. Some other optimization algorithms were merged with the DGA to increase the fault diagnostic accuracy, such as Teaching learning-based optimization [30], adaptive dynamic meta-heuristics [2], and Grey Wolf Optimization [12].

This paper uses a novel Meta-heuristic algorithm, based on the Gravitational Search and Dipper Throated Optimization Algorithms (GSDTO), to enhance the diagnostic accuracy of the transformer faults' misinterpretation with the other techniques, whether the traditional DGA or that merged the AI with the traditional DGA methods. As the Dipper Throated Optimization (DTO) algorithm relies on numerous variables in the optimization process, its performance degrades. In addition, the convergence of the algorithm is premature. However, the satisfactory balance between exploration and exploitation is a considerable advantage. To take advantage of this benefit, the suggested approach employs the DTO algorithm. Gravitational Search Algorithm (GSA), despite its simplicity and outstanding balance between exploration and exploitation, has disadvantages such as a low exploration rate and performance decrease when a large number of local optimum solutions exist. Utilizing the dipper throated optimizer, this study uses this algorithm to take advantage of its benefits while compensating for its limitations.

The proposed model uses the raw gases' concentration regardless of the gases' ratios considered in the conventional DGA techniques, which resulted in a wrong diagnosis of the fault due to the out-of-code cases. ANOVA statistical analysis is applied to investigate the stability of the proposed model with the uncertainty that occurred in the input data. The results and the statistical analysis indicated the robustness of the constructed algorithm and enhanced the diagnostic accuracy of the transformer faults for the testing samples. The dataset employed in this work consists of 460 samples collected from the literature and the Egyptian Electricity Holding Company central chemical laboratory.

The proposed GSDTO enhances the Long Short Term Memory (LSTM) classification method parameters. Excellent diagnostic accuracy of the transformer faults is achieved based on the proposed GSDTO+LSTM classification model. A binary version of the proposed (GSDTO) algorithm is first used for feature selection from the tested dataset. The binary GSDTO (bGSDTO) algorithm is tested first compared to PSO [31], Grey Wolf Optimizer (GWO) [32], Whale Optimization Algorithm (WOA) [33], Biogeography-Based Optimizer (BBO) [34], Firefly Algorithm (FA) [35], Genetic Algorithm (GA) [36], and Bat Algorithm (BA) [37]. Then, a classifier based on the suggested GSDTO algorithm and LSTM method is examined for the tested dataset. Comparative analysis is performed between the GS-DTO+LSTM algorithms and WOA+LSTM, GWO+LSTM, GA+LSTM, and PSO+LSTM algorithms. The GSDTO+LSTM algorithm's diagnostic accuracy is also examined using randomly selected data from the total number of samples.

This work's main contributions can be expressed as follows:

- A novel Gravitational Search Dipper Throated Optimization Algorithm (GSDTO) is proposed.
- A binary GSDTO algorithm, a binary version of the proposed algorithm, is applied for feature selection from the tested dataset.
- A GSDTO+LSTM classifier, based on the proposed GSDTO algorithm and LSTM method, is developed to improve the tested dataset classification accuracy.
- The GSDTO algorithm's statistical difference is tested by Wilcoxon's rank-sum and ANOVA tests.
- The GSDTO algorithm is used to improve the LSTM classification method performance for classifying purposes which can be applied in a new high voltage engineering application.
- The binary GSDTO algorithm and the LSTM-based classification algorithm can be generalized and tested for various types of datasets.

The organization of this paper is as follows. Materials and methods as presented in Section 2. Section 3 discusses the proposed GSDTO algorithm, binary GSDTO algorithm, and the GSDTO+LSTM-based model. Section 4 shows the experimental results and analysis. The validation and discussion of the proposed GSDTO-based model compared to state-of-the-art models are discussed in this section. In Section 6, the conclusion of this work and the future directions are presented.

2. Materials and Methods

2.1. Distribution of the Data

A total number of 460 samples were used in this study. A number of 386 samples for training and 74 samples for testing purposes were used to investigate the robustness of the constructed model. The collection of samples was from the central chemical laboratory of Egyptian Electricity Holding Company and literature. The dataset has several attributes including Hydrogen (H_2), Methane (CH_4), Ethane (C_2H_6), Ethylene (C_2H_4), Acetylene (C_2H_2), Power factor, Interfacial V, Dielectric rigidity, Water content, Health index, Life expectation, Status, Category (ACT), and Time. Table 1 shows the distribution of training data samples. The data samples in the testing process are categorized as 43 samples for Partial Discharge (PD), 69 samples for low energy discharge (D1), 115 samples for high energy discharge (D2), 81 samples for a low thermal fault (T1), 24 samples for a medium thermal fault (T2), and 54 samples for a high thermal fault (T3). Table 2 illustrates the distribution of testing samples based on the source of the data and the fault types. The data samples in the testing process are categorized as seven samples for PD, 13 samples for D1, 24 samples for D2, 16 samples for T1, 4 samples for T2, and 10 samples for T3.

Ref.	PD	D1	D2	T1	T2	Т3	Total
[8]	2	0	0	3	0	0	5
[9]	9	24	48	0	0	18	99
[25]	0	2	1	1	3	1	8
[38]	27	42	55	70	18	28	240
[39]	1	0	5	2	0	1	9
[40]	3	0	4	4	3	5	19
[41]	1	1	2	1	0	1	6
Total	43	69	115	81	24	54	386

Table 1. Distribution of the 386 training samples according to the fault types and the references.

Table 2. The distribution of the 74 testing samples according to the fault types and the references.

Ref.	PD	D1	D2	T1	T2	Т3	Total
[9]	1	6	8	1		1	17
[38]	6	6	11	11	3	2	39
[40]			2	1			3
[41]						1	1
[42]						2	2
[43]			1				1
[44]		1	1	1			3
[45]						1	1
[46]				2	1	3	6
[47]			1				1
Total	7	13	24	16	4	10	74

2.2. Machine Learning

One type of classification and prediction model is an artificial neural network (ANN). Complex interactions between data patterns or sets of cause-and-effect variables are modeled with artificial neural networks (ANN). These techniques may include transient detection and pattern recognition. ANN is an information processing pattern composed of neurons that work together to solve problems similar to the brain. When developing an algorithmic solution and extracting the structure from existing data, a neural network is useful [48].

Random forest, a technique based on statistical learning theory, offers a number of benefits, including fewer configurable parameters, greater prediction precision, and increased generalization capacity. It takes several samples from the original sample using the bootstrap sampling method, constructs decision tree models based on each bootstrap sample, integrates multiple decision tree predictions, and determines the outcome through a voting procedure [48].

The conclusions of the prediction algorithm utilizing the k-NN technique are based on historical events similar to the current state based on a measure of distance. The k-NN output values, simple average or weighted averaging, are used to generate forecasts. Consequently, specialists can evaluate the k-NN method's outcomes. In k-NN numerical prediction, the object's predictable variable is the average value of its k nearest neighbors. The k-NN algorithm is a fundamental and effective tool for machine learning [48].

The LSTM model is an improved artificial neural network model that may be used to solve a variety of problems, as detailed in [49]. The primary advantage of the LSTM is its ability to retain information for an extended period of time. Figure 1 illustrates the LSTM design. The LSTM model's initial step is to determine which data from the cell state should be disregarded. As seen in Equation (1), this is accomplished using a forget gate or sigmoid layer.

$$f_t = \sigma(b_f + W_f[h_{t-1}, x_t]) \tag{1}$$



Figure 1. Architecture of LSTM Neural Network.

The new data to be stored in the cell state should be determined. The sigmoid layer determines which values require updating, and the tanh layer adds the new candidate to the produced state, as described in Equations (2) and (3).

$$i_t = \sigma(b_i + W_i[h_{t-1}, x_t]) \tag{2}$$

$$C'_{t} = \tanh(b_{i} + W_{i}[h_{t-1}, x_{t}])$$
(3)

The old cell state, C_{t-1} , is updated into the new cell state, C_t , based on the following equation using Equations (1)–(3).

$$C_t = i_t \times C'_t + f_t \times C_{t-1} \tag{4}$$

The cell state will then multiply the sigmoid gate output by tanh and force values between [-1,1].

$$h_t = o_t \times \tanh(C_t), o_t = \sigma(b_o + W_o[h_{t-1}, x_t])$$
(5)

2.3. Dipper Throated Optimization (DTO)

Among passerines, dipper throated birds are unusual. They are excellent swimmers, divers, and hunt underwater. Additionally, their flexible and small wings enable them to fly straight and fast without glides or pauses. The algorithm based on the dipper throated birds' (DTO) behavior assumes that birds fly and swim in search of food supplies, with N_{fs} representing the number of birds. Birds' positions are marked by **BP**, while **BV** denotes their velocities. The parameters **BP** and **BV** are denoted as follows [50]:

$$\mathbf{BP} = \begin{bmatrix} BP_{1,1} & BP_{1,2} & BP_{1,3} & \dots & BP_{1,d} \\ BP_{2,1} & BP_{2,2} & BP_{2,3} & \dots & BP_{2,d} \\ BP_{3,1} & BP_{3,2} & BP_{3,3} & \dots & BP_{3,d} \\ \dots & \dots & \dots & \dots & \dots \\ BP_{n,1} & BP_{n,2} & BP_{n,3} & \dots & BP_{n,d} \end{bmatrix}$$
(6)
$$\mathbf{BV} = \begin{bmatrix} BV_{1,1} & BV_{1,2} & BV_{1,3} & \dots & BV_{1,d} \\ BV_{2,1} & BV_{2,2} & BV_{2,3} & \dots & BV_{2,d} \\ BV_{3,1} & BV_{3,2} & BV_{3,3} & \dots & BV_{3,d} \\ \dots & \dots & \dots & \dots & \dots \\ BV_{n,1} & BV_{n,2} & BV_{n,3} & \dots & BV_{n,d} \end{bmatrix}$$
(7)

where $BP_{i,j}$ denotes *i*th position of bird in the *j*th dimension and $BV_{i,j}$ denotes *i*th bird velocity in the *j*th dimension. The initial locations of $BP_{i,j}$ are distributed uniformly between

the lower and higher boundaries. The fitness values f_n are calculated as shown in the accompanying array.

$$\mathbf{f} = \begin{bmatrix} \mathbf{f}_{1}(BP_{1,1}, BP_{1,2}, BP_{1,3}, \dots, BP_{1,d}) \\ \mathbf{f}_{2}(BP_{2,1}, BP_{2,2}, BP_{2,3}, \dots, BP_{2,d}) \\ \mathbf{f}_{3}(BP_{3,1}, BP_{3,2}, BP_{3,3}, \dots, BP_{3,d}) \\ \dots \\ \mathbf{f}_{n}(BP_{n,1}, BP_{n,2}, BP_{n,3}, \dots, BP_{n,d}) \end{bmatrix}$$
(8)

Following that, these values are arranged ascendingly. \mathbf{BP}_{best} is declared as the first best solution. The remainder of the answers is expected to be regular birds \mathbf{BP}_{nd} for follower birds. \mathbf{BP}_{Gbest} is declared to be the global best solution. The term *R* is a random value within [0, 1] and is employed in the DTO algorithm to switch between swimming and flying birds. If *R* < 0.5, the birds are considered to be swimming birds, and their positions are changed as

$$\mathbf{BP}_{nd}(t+1) = \mathbf{BP}_{best}(t) - \mathbf{C}_1 \cdot |\mathbf{C}_2 \cdot \mathbf{BP}_{best}(t) - \mathbf{BP}_{nd}(t)|$$
(9)

where $\mathbf{BP}_{best}(t)$ is the best bird's position at iteration *t* and $\mathbf{BP}_{nd}(t)$ is a bird's position. The parameters of \mathbf{C}_1 and \mathbf{C}_2 are calculated as follows.

$$C_1 = 2C \cdot \mathbf{r}_1 - C,$$

$$C_2 = 2\mathbf{r}_1,$$

$$C = 2\left(1 - \left(\frac{t}{T_{max}}\right)^2\right)$$
(10)

where *C* changes exponentially from 2 to 0, parameter of \mathbf{r}_1 has a random value within [0,1] and maximum number of iterations is denoted as T_{max} .

If $R \ge 0.5$, the birds are considered to be flying birds and their positions are updated as follows:

$$\mathbf{BP}_{nd}(t+1) = \mathbf{BP}_{nd}(t) + \mathbf{BV}(t+1)$$
(11)

The updated bird's velocity, denoted as $\mathbf{BV}(t+1)$, is calculated as

$$\mathbf{BV}(t+1) = \mathbf{C}_{3}\mathbf{BV}(t) + \mathbf{C}_{4}\mathbf{r}_{2}(\mathbf{BP}_{best}(t) - \mathbf{BP}_{nd}(t)) + \mathbf{C}_{5}\mathbf{r}_{2}(\mathbf{BP}_{Gbest} - \mathbf{BP}_{nd}(t))$$
(12)

where C_3 is a weight value, C_4 and C_5 are constants. **BP**_{*Gbest*} represents global best position (best of all agents in the population) and r_2 parameter value is within [0;1]. The DTO algorithm is explained step by step in Algorithm 1 [50].

2.4. Gravitational Search Algorithm (GSA)

The GSA algorithm was proposed based on Newton's laws of motion and gravity and was applied for Solar Radiation Forecasting in [51]. The position and gravitational masses (inertial, active, and passive) represent the main attributes for each agent in this algorithm. A problem's solution is described by these attributes and decided by a fitness function. In GSA, the agent's position is defined as in the following equation.

$$\mathbf{x}_i = \left(x_i^1, \dots, x_i^d, \dots, x_i^N\right), i = 1, 2, \dots, n$$
(13)

where x_i^d indicates *i*th agent position in d^{th} dimension of *N* dimensions and *n* is number of agents (masses). The $\mathbf{x}_i^d(t)$, the position of each agent, is updated as

$$\mathbf{x}_{i}^{d}(t+1) = \mathbf{x}_{i}^{d}(t) + \mathbf{v}_{i}^{d}(t+1),$$
(14)

where $\mathbf{v}_{i}^{d}(t)$, the velocity of an agent, is calculated as

$$\mathbf{v}_i^d(t+1) = \mathbf{rand}_i \cdot \mathbf{v}_i^d(t) + \mathbf{a}_i^d(t)$$
(15)

where **rand**_{*i*} is a random value in [0, 1]. The acceleration of agent *i* in d^{th} direction, denoted as $\mathbf{a}_i^d(t)$, is updated using the *i*th agent inertial mass, denoted as $\mathbf{M}_{ii}(t)$, as

$$\mathbf{a}_{i}^{d}(t) = \frac{\mathbf{F}_{i}^{d}(t)}{\mathbf{M}_{ii}(t)}$$
(16)

Algorithm 1 DTO Algorithm.

1: **Initialization** positions of agents **BP**_{*i*}(i = 1, 2, ..., n) with *n* agents, velocities of agents \mathbf{BV}_i (*i* = 1, 2, ..., *n*), maximum iterations T_{max} , objective function \mathbf{f}_n , parameters of \mathbf{r}_1 , $\mathbf{r}_2, \mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, \mathbf{C}_4, \mathbf{C}_5, C, R, t = 1$ 2: **Obtain f**_{*n*} for each agent **BP**_{*i*} 3: Find best agent BP_{best} 4: while $t \leq T_{max}$ do for (i = 1 : i < n + 1) do 5: **if** (*R* < 0.5) **then** 6: 7: Update current swimming agent position as $\mathbf{BP}_{nd}(t+1) = \mathbf{BP}_{best}(t) - \mathbf{C}_1 |\mathbf{C}_2 \cdot \mathbf{BP}_{best}(t) - \mathbf{BP}_{nd}(t)|$ 8: else Update current flying agent velocity as 9: $\mathbf{BV}(t+1) = \mathbf{C}_{3}\mathbf{BV}(t) + \mathbf{C}_{4}\mathbf{r}_{2}(\mathbf{BP}_{best}(t) - \mathbf{BP}_{nd}(t)) + \mathbf{C}_{5}\mathbf{r}_{2}(\mathbf{BP}_{Gbest} - \mathbf{BP}_{nd}(t))$ Update current flying agent position as 10: $\mathbf{BP}_{nd}(t+1) = \mathbf{BP}_{nd}(t) + \mathbf{BV}(t+1)$ end if 11: end for 12: **Obtain f**_{*n*} for each agent **BP**_{*i*} 13: 14: **Update C**₁, **C**₂, *C*, *R* 15: Find best agent **BP**_{best} **Set BP**_{Gbest} = **BP**_{best} 16: 17: **Set** *t* = *t* + 1 18: end while 19: Return BP_{Gbest}

Total gravitational force, denoted as $\mathbf{F}_{i}^{d}(t)$, is calculated as

$$\mathbf{F}_{i}^{d}(t) = \sum_{j=1, j \neq i}^{N} \operatorname{rand}_{j} \mathbf{F}_{ij}^{d}(t)$$
(17)

where **rand**_{*j*} is a random value in [0, 1]. The force acting on mass *i* from mass *j*, denoted as $\mathbf{F}_{ii}^{d}(t)$, is updated by the following equation.

$$\mathbf{F}_{ij}^{d}(t) = \mathbf{G}(t) \frac{\mathbf{M}_{pi}(t) \times \mathbf{M}_{aj}(t)}{||\mathbf{x}_{i}(t), \mathbf{x}_{j}(t)||_{2} + \varepsilon} (\mathbf{x}_{j}^{d}(t) - \mathbf{x}_{i}^{d}(d))$$
(18)

where \mathbf{M}_{pi} represents passive gravitational mass for agent *i* and \mathbf{M}_{aj} represents active gravitational mass for agent *j*. The $\mathbf{G}(t)$ parameter indicates the gravitational constant at time *t* and ε represents a constant value. The term $||\mathbf{x}_i(t), \mathbf{x}_j(t)||_2$ denotes Euclidean distance between agent *i* and agent *j*.

The fitness evaluation computes gravitational and inertia masses. The values of masses are computed using the fitness map, assuming that the gravitational and iner-

tia mass are identical. The gravitational and inertial masses are revised by using the following equations:

$$\mathbf{M}_{pi} = \mathbf{M}_{aj} = \mathbf{M}_{ii} = \mathbf{M}_{i}, \ i = 1, 2, \dots, N$$
(19)

$$\mathbf{M}_{i}(t) = \frac{\mathbf{m}_{i}(t)}{\sum_{i=1}^{N} \mathbf{m}_{i}(t)}, \ \mathbf{m}_{i}(t) = \frac{\mathbf{fit}_{i}(t) - worst(t)}{best(t) - worst(t)}$$
(20)

where $\mathbf{fit}_i(t)$ indicates the agent *i* fitness value at iteration t. For a minimization problem, worst(t) and best(t) are calculated as in the following equations.

$$best(t) = \min_{j \in \{1,\dots,N\}} \mathbf{fit}_j(t)$$
(21)

$$worst(t) = \max_{j \in \{1, \dots, N\}} \mathbf{fit}_j(t)$$
(22)

The GSA algorithm is explained step by step in Algorithm 2. Since all possible solutions are used to update the position of each solution, the GSA algorithm exhibits an extremely exploratory nature. Each solution in a rotation can affect the others depending on their distances and quality. However, the precision of this approach is frequently suboptimal.

Algorithm 2 GSA Algorithm.

- 1: **Initialization** positions of agents \mathbf{x}_i (i = 1, 2, ..., n) with n agents, maximum iterations T_{max} , objective function \mathbf{f}_n , parameters of \mathbf{rand}_i , \mathbf{rand}_i , t = 1
- 2: **Obtain** \mathbf{f}_n for each agent \mathbf{x}_i
- 3: **Find** best agent **x**_{best}
- 4: while $t \leq T_{max}$ do
- 5: **for** (i = 1 : i < n + 1) **do**
- 6: **Update** gravitational and inertia masses by Equations (19) and (20)
- 7: **Update** acceleration of current agent by

$$\mathbf{a}_{i}^{d}(t) = rac{\mathbf{F}_{i}^{d}(t)}{\mathbf{M}_{ii}(t)}$$

Update velocity of current agent by $\mathbf{v}_i^d(t+1) = \operatorname{rand}_i \cdot \mathbf{v}_i^d(t) + \mathbf{a}_i^d(t)$

- 9: **Update** position of current agent by
- $\mathbf{x}_i^d(t+1) = \mathbf{x}_i^d(t) + \mathbf{v}_i^d(t+1)$
- 10: **end for**

8:

- 11: **Obtain** \mathbf{f}_n for each agent \mathbf{x}_i
- 12: **Update rand**_{*i*}, rand_{*j*}, t = t + 1
- 13: **Find** best agent **x**_{best}
- 14: **Set** $\mathbf{x}_{Gbest} = \mathbf{x}_{best}$
- 15: end while
- 16: Return x_{Ghest}

3. Proposed Methodology

3.1. Proposed GSDTO Algorithm

The proposed Gravitational Search Dipper Throated optimization (GSDTO) algorithm is presented in Algorithm 3 step by step. The GSDTO algorithm covers the disadvantages of the DTO and GSA algorithms and combines their advantages to achieve the best global solution. The algorithm starts with initializing the positions of predetermined *n* agents \mathbf{x}_i (i = 1, 2, ..., n) and their velocities \mathbf{v}_i (i = 1, 2, ..., n). It sets the allowed iterations for the execution process as T_{max} , the objective function \mathbf{f}_n , the DTO parameters of \mathbf{r}_1 , \mathbf{r}_2 , \mathbf{C}_1 , \mathbf{C}_2 , \mathbf{C}_3 , \mathbf{C}_4 , \mathbf{C}_5 , *C*, *R*, and the GSA parameters of \mathbf{rand}_i and \mathbf{rand}_j . The term $rand_{GSDTO}$ is a random value within [0, 1].

Algorithm 3 Proposed GSDTO Algorithm.

1: **Initialization** positions of agents \mathbf{x}_i (i = 1, 2, ..., n) with *n* agents, velocities of agents \mathbf{v}_i (*i* = 1, 2, ..., *n*), maximum number of iterations T_{max} , objective function \mathbf{f}_n , parameters of \mathbf{r}_1 , \mathbf{r}_2 , \mathbf{C}_1 , \mathbf{C}_2 , \mathbf{C}_3 , \mathbf{C}_4 , \mathbf{C}_5 , C, R, rand_i, rand_j, rand_{GSDTO}, t = 12: **Obtain** \mathbf{f}_n for each agent \mathbf{x}_i 3: **Find** best agent **x**_{best} 4: Set $\mathbf{x}_{Gbest} = \mathbf{x}_{best}$ 5: while $t \leq Tmax$ do if $(rand_{GSDTO} > 0.5)$ then 6: for (i = 1 : i < n + 1) do 7: **if** (*R* < 0.5) **then** 8: **Update** current swimming agent position by 9: $\mathbf{x}(t+1) = \mathbf{x}_{best}(t) - \mathbf{C}_1 |\mathbf{C}_2 \cdot \mathbf{x}_{best}(t) - \mathbf{x}(t)|$ else 10: **Update** current flying agent velocity by 11: $\mathbf{v}(t+1) = \mathbf{C}_3 \mathbf{v}(t) + \mathbf{C}_4 \mathbf{r}_2 (\mathbf{x}_{best}(t) - \mathbf{x}(t)) + \mathbf{C}_5 \mathbf{r}_2 (\mathbf{x}_{Gbest} - \mathbf{x}(t))$ Update current flying agent position by 12: $\mathbf{x}(t+1) = \mathbf{x}(t) + \mathbf{v}(t+1)$ 13: end if end for 14: else 15: for (i = 1 : i < n + 1) do 16: 17: **Update** gravitational and inertia masses by Equations (29) and (30) $\mathbf{F}(t)$ **Update** acceleration of current agent by $\mathbf{a}(t) =$ 18: $\mathbf{M}_{ii}(t)$ **Update** velocity of current agent by Equation (27) 19: **Update** position of current agent by Equation (26) 20: 21: end for 22: end if **Obtain** \mathbf{f}_n for each agent \mathbf{x}_i 23: Update C_1 , C_2 , C, R, rand_i, rand_i, rand_{GSDTO} 24: 25: Find best agent **x**_{best} 26: Set $\mathbf{x}_{Gbest} = \mathbf{x}_{best}$ 27: **Set** t = t + 128: end while 29: **Return** best agent **x**_{*Gbest*}

If $rand_{GSDTO} > 0.5$, the GSDTO algorithm starts to update the positions and velocities of agents as in the following equations. The positions will be updated considering swimming agent if R < 0.5 by

$$\mathbf{x}(t+1) = \mathbf{x}_{best}(t) - \mathbf{C}_1 \cdot |\mathbf{C}_2 \cdot \mathbf{x}_{best}(t) - \mathbf{x}(t)|$$
(23)

Otherwise, agents will be considered as flying agents and the positions is updated as

$$\mathbf{x}(t+1) = \mathbf{x}(t) + \mathbf{v}(t+1)$$
(24)

where $\mathbf{v}(t+1)$, updated velocity of each agent, will be calculated as

$$\mathbf{v}(t+1) = \mathbf{C}_3 \mathbf{v}(t) + \mathbf{C}_4 \mathbf{r}_2(\mathbf{x}_{best}(t) - \mathbf{x}_{nd}(t)) + \mathbf{C}_5 \mathbf{r}_2(\mathbf{x}_{Gbest} - \mathbf{x}(t))$$
(25)

If $rand_{GSDTO} \le 0.5$, the GSDTO algorithm will update the positions and velocities of the agents according to the following equations. The $\mathbf{x}(t)$, position of each agent at iteration t, will be calculated as follows.

$$\mathbf{x}(t+1) = \mathbf{x}(t) + \mathbf{v}(t+1), \tag{26}$$

where $\mathbf{v}(t)$, velocity of each agent at iteration t, is changed as

$$\mathbf{v}(t+1) = \operatorname{rand}_{i} \cdot \mathbf{v}(t) + \mathbf{a}(t)$$
(27)

where the parameter $\mathbf{a}(t)$, acceleration of each agent at iteration t, changes as

$$\mathbf{a}(t) = \frac{\mathbf{F}(t)}{\mathbf{M}_{ii}(t)}$$
(28)

The values of masses are computed using the fitness map for a minimization problem, assuming that the gravitational and inertia mass are identical by the following equations:

$$\mathbf{M}_{pi} = \mathbf{M}_{ai} = \mathbf{M}_{ii} = \mathbf{M}_i, \ i = 1, 2, \dots, N$$
⁽²⁹⁾

$$\mathbf{M}_{i}(t) = \frac{\mathbf{m}_{i}(t)}{\sum_{j=1}^{N} \mathbf{m}_{j}(t)}$$
(30)

$$\mathbf{m}_{i}(t) = \frac{\mathbf{fit}_{i}(t) - \max_{j \in \{1,\dots,N\}} \mathbf{fit}_{j}(t)}{\min_{j \in \{1,\dots,N\}} \mathbf{fit}_{j}(t) - \max_{j \in \{1,\dots,N\}} \mathbf{fit}_{j}(t)}$$
(31)

where $fit_i(t)$ indicates the agent *i* fitness value at iteration *t*.

The GSDTO algorithm's computational complexity in this work is expressed as follows. For iterations T_{max} and n number of agents, the complexity is defined as

- Initialize parameters of the GSDTO algorithm, \mathbf{x}_i (i = 1, 2, ..., n), \mathbf{v}_i (i = 1, 2, ..., n), T_{max} , \mathbf{C}_1 , \mathbf{C}_2 , \mathbf{C}_3 , \mathbf{C}_4 , \mathbf{C}_5 , \mathbf{r}_1 , \mathbf{r}_2 , C, R, rand_i, rand_j, rand_{GSDTO}, and t = 1: O(1).
- Calculate \mathbf{f}_n for each agent \mathbf{x}_i : O(n).
- Obtain the best agent \mathbf{x}_{best} : O(n).
- Update current swimming agent position: $O(T_{max} \times n)$.
- Update current flying agent velocity: $O(T_{max} \times n)$.
- Update current flying agent position: $O(T_{max} \times n)$.
- Update acceleration of current agent $\mathbf{a}_i(t)$: $O(T_{max} \times n)$.
- Update velocity of current agent $\mathbf{v}_i(t+1)$: $O(T_{max} \times n)$.
- Update position of current agent $\mathbf{x}_i(t+1)$: $O(T_{max} \times n)$.
- Calculate \mathbf{f}_n for each agent \mathbf{x}_i : $O(T_{max})$.
- Update C_1 , C_2 , C, R, rand_i, rand_j, rand_{GSDTO}: $O(T_{max})$.
- Obtain best agent \mathbf{x}_{best} : $O(T_{max})$.
- Set $\mathbf{x}_{Gbest} = \mathbf{x}_{best}$: $O(T_{max})$.
- Set t = t + 1: $O(T_{max})$.
- Obtain global best agent \mathbf{x}_{Gbest} : O(1)

Based on the above analysis of the GSDTO algorithm, the computation complexity is set to $O(T_{max} \times n)$ and it will be $O(T_{max} \times n \times d)$ for *d* dimension.

3.2. Proposed Binary GSDTO Algorithm

The solutions of the GSDTO algorithm will be strictly binary, with values of 0 or 1, in case of feature selection issues. Thus, the continuous values of the proposed GSDTO algorithm will be transformed to binary [0, 1] to facilitate the feature selection process from the dataset. This study employs the following equation, which is based on the *Sigmoid* function [52].

$$x_d^{t+1} = \begin{cases} 1 & \text{if } Sigmoid(m) \ge 0.5\\ 0 & otherwise \end{cases},$$

$$Sigmoid(m) = \frac{1}{1 + e^{-10(m-0.5)}},$$
(32)

where x_d^{t+1} , at iteration t and dimension d, denotes the binary solution. The *Sigmoid* function can scale the output solutions to be binary ones. If $Sigmoid(m) \ge 0.5$, this will change the value to 1; otherwise, the value will be 0. The m parameter reflects the algorithm's selected features. Figure 2 shows how the *Sigmoid* function scales the output solutions to binary [0, 1]. The binary GSDTO algorithm is described in detail in Algorithm 4. By analyzing the bGSDTO algorithm, the computation complexity is set to $O(t_{max} \times n)$ and it will be $O(t_{max} \times n \times d)$ for d dimension.



Figure 2. Sigmoid function to scale the output solutions to binary [0, 1].

Algorithm 4 Proposed Binary GSDTO Algorithm.

1: Initialization parameters of the GSDTO 2: **Convert** solution to binary [0 or 1] Obtain fitness for each agent and Find best agent 3: while $t \leq Tmax$ do 4: 5: if $(rand_{GSDTO} > 0.5)$ then for (i = 1 : i < n + 1) do 6: **if** (R < 0.5) **then** 7: Update current swimming agent position 8: 9: else 10: Update current flying agent velocity and position 11: end if end for 12: 13: else for (i = 1 : i < n + 1) do 14: Update gravitational and inertia masses 15: Update acceleration of current agent 16: 17: Update velocity and position of current agent end for 18: 19: end if 20: **Convert** updated solution to binary 21: **Obtain** fitness for each agent and Find best agent Update parameters 22: 23: end while 24: Return best solution

3.3. Proposed GSDTO+LSTM Based Model

Figure 3 shows the proposed model based on the presented GSDTO algorithm. The proposed GSDTO+LSTM-based model is constructed based on two main phases: feature

selection and classification. The first phase of the proposed model focuses on preprocessing and feature selection. The preprocessing is interesting in cleaning and normalizing the tested dataset, including correlation analysis, feature scaling, and removing null values. Feature selection in this phase is performed after cleaning the dataset. The binary version of the proposed GSDTO algorithm is employed in this step for selecting the best number of features from the total number of attributes in the tested dataset. The validation of this phase, to confirm the quality of the bGSDTO algorithm in feature selection, is performed in the experiments using feature selection performance metrics, including average error and standard deviation.



Figure 3. Proposed feature selection and classification model based on the proposed GSDTO algorithm.

The second phase, the intermediate phase in Figure 3, which is utilized to compare results, includes applying the base models of NN, k-NN, and RF for the selected features in phase one. The output results of the three base classifiers in phase two are recorded to be compared with the GSDTO+LSTM model classification results in the third phase. The last stage of the proposed GSDTO+LSTM-based model, the classification phase, includes optimizing the LSTM model parameters by the presented GSDTO algorithm. The classification in the third phase of the model is applied based on the selected features in phase one, and the output results are compared with the results of phase two as the final results of this model.

4. Experimental Results

This section explains the results of this study in detail. The experiments are divided into three scenarios. The first scenario discusses the feature selection ability of the proposed bGSDTO algorithm for the tested dataset, while the second scenario shows the presented algorithm's ability for classification purposes. In the third and last scenario, the GSDTO+LSTM algorithm's diagnostic accuracy is examined using randomly selected data from the total number of samples.

4.1. Feature Selection Scenario

The binary version of the proposed (GSDTO) algorithm is used for feature selection from the tested dataset as mentioned in Figure 3. In the first scenario, the feature selection results of this study's presented GSDTO algorithm are discussed. Table 3 shows the GSDTO algorithm configuration of all parameters used in the experiment, while Table 4 presents the compared algorithms configuration. The binary GSDTO (bGSDTO) algorithm is tested compared to PSO [31], GWO [32], WOA [33], BBO [34], FA [35], GA [36], and BA [37].

Parameter (s)	Value (s)
# Agents	10
# Iterations	80
# Repetitions	20
Dimension	# features
С	[0,2]
R	[0,1]
α of F_n	0.99
β of F_n	0.01

Table 3. Configuration parameters of the GSDTO algorithm.

 Table 4. Configuration parameters of the compared algorithms.

Algorithm	Parameter (s)	Value (s)
PSO	W _{max} , W _{min}	[0.9, 0.6]
	<i>C</i> ₁ , <i>C</i> ₂	[2, 2]
GWO	a	2 to 0
WOA	a	2 to 0
	r	[0, 1]
BBO	Habitat modification Probability	1.0
	Mutation Probability	0.05
	Immigration Probability	[0, 1]
	Migration rate	1.0
	Max immigration	1.0
	Step size	1.0
BA	Pluse rate	0.5
	Loudness	0.5
	Frequency	[0, 1]
GA	Crossover	0.9
	Mutation ratio	0.1
	Mechanism of Selection	Roulette wheel
FA	# Fireflies	10

The objective equation, \mathbf{f}_n , is used to determine the quality of a solution in the binary GSDTO algorithm. \mathbf{f}_n is employed in the following equation for a classifier's error rate, *Err*, a number of selected features, *s*, and a number of missing features, *S*, as follows.

$$\mathbf{f}_n = \alpha Err + \beta \frac{|s|}{|S|} \tag{33}$$

where $\alpha \in [0, 1]$ and $\beta = 1 - \alpha$ indicate the population relevance of the given feature. The approach is satisfactory if it can provide a subset of features capable of producing a low classification error rate. K-nearest neighbor (k-NN) is a widely used, straightforward classification technique. The goodness of the chosen features is ensured in this method by using the k-NN as a classifier. The only factor used to determine classifiers is the smallest distance between the query instance and the training examples; no model for K-nearest neighbors is used in this experiment.

In Table 5, the performance metrics employed to measure the feature selection results of the proposed algorithm are presented. The variables mentioned in Table 5 are: number of runs of the optimizer, M, best solution at run j, g_j^* , the g_j^* vector size, $size(g_j^*)$, number of features, D, and N as number points, the output label of the classifier for a point i, C_i , the label of class for a point i, L_i , and the *Match* function for matching calculation between two inputs.

Table 5. Feature selection performance metrics.

Metric	Value
Average Error	$1 - \frac{1}{M}\sum_{j=1}^{M} \frac{1}{N}\sum_{i=1}^{N} Match(C_i, L_i)$
Average Select Size	$rac{1}{M} \sum_{j=1}^{M} rac{size(g_j^*)}{D}$
Average Fitness	$\frac{1}{M}\sum_{i=1}^{M}g_{i}^{*}$
Best Fitness	$Min_{i=1}^{M}g_{i}^{*}$
Worst Fitness	$Max_{j=1}^{M}g_{j}^{*}$
Standard Deviation	$\sqrt{rac{1}{M-1}\sum(g_j^*-Mean)^2}$

The suggested and compared algorithms' feature selection results, based on 20 runs and 80 iterations for ten agents as mentioned in Table 3, are provided in Table 6. The minimum average error of (0.1969) and the standard deviation of (0.0824) show the performance of the presented bGSDTO algorithm. The second best algorithm in feature selection of the tested data, minimum average error, is bGWO with (0.2141), followed by bBBO with (0.2161) and then bGA with (0.2277). The worst algorithm in feature selection is the bBA algorithm, with an average error of (0.2575). The selected features by the proposed algorithm from the tested dataset total features are named (H_2), (CH_4), (C_2H_6), (C_2H_4), (C_2H_2), and (ACT).

Table 6. Presented bGSDTO and compared algorithms feature selection results.

	bGSDTO	bGWO	bPSO	bBA	bWOA	bBBO	bFA	bGA
Average error	0.1969	0.2141	0.2479	0.2575	0.2477	0.2161	0.2463	0.2277
Average Select size	0.1497	0.3497	0.3497	0.4891	0.5131	0.5135	0.3842	0.2921
Average Fitness	0.2601	0.2763	0.2747	0.2976	0.2825	0.2804	0.3266	0.2877
Best Fitness	0.1619	0.1966	0.255	0.1873	0.2466	0.2701	0.2453	0.191
Worst Fitness	0.2604	0.2635	0.3227	0.2889	0.3227	0.3566	0.3429	0.3061
Standard deviation Fitness	0.0824	0.0871	0.0865	0.0964	0.0887	0.1314	0.1233	0.0887

The statistical analysis, using a one-way analysis of variance (ANOVA) and Wilcoxon Signed Rank tests, is conducted to determine the performance of the proposed binary GSDTO algorithm based on the average error. Wilcoxon's test is utilized to determine the *p*-values comparing the suggested algorithm to other algorithms. This statistical test can assess whether or not there is a significant difference between the results of the proposed algorithm and those of other algorithms with a *p*-value < 0.05. To determine whether there is a statistically significant difference between the suggested algorithm and other algorithms analyzed, an ANOVA test was also conducted. The ANOVA test results for the suggested versus compared algorithms are shown in Table 7, while Table 8 compares the proposed and compared algorithms using the Wilcoxon Signed-Rank test. The statistical analysis is performed using 20 runs of the presented and compared algorithms for fair comparisons.

Table 7. Presented bGSDTO versus compared algorithms ANOVA test results.

	SS	DF	MS	F (DFn, DFd)	<i>p</i> Value
Treatment (between columns)	0.06262	7	0.008946	F (7, 152) = 173.9	p < 0.0001
Residual (within columns)	0.007819	152	0.00005144	-	-
Total	0.07044	159	-	-	-

Table 8. Presented bGSDTO and com	pared algorithms Wilco	xon Signed-Rank test results.
	0	0

	bGSDTO	bGWO	bPSO	bBA	bWOA	bBBO	bFA	bGA
Theoretical median	0	0	0	0	0	0	0	0
Actual median	0.1969	0.2141	0.2479	0.2575	0.2477	0.2161	0.2463	0.2277
# Values	20	20	20	20	20	20	20	20
Wilcoxon Signed Rank Test								
Signed ranks' sum	210	210	210	210	210	210	210	210
Positive ranks' sum	210	210	210	210	210	210	210	210
Negative ranks' sum	0	0	0	0	0	0	0	0
P value (two tailed)	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001
Exact or estimate?	Exact							
Significant ($alpha = 0.05$)?	Yes							
Discrepancy	0.1969	0.2141	0.2479	0.2575	0.2477	0.2161	0.2463	0.2277

Figure 4 uses the convergence curve for comparing the provided and comparable algorithms in feature selection. During the population initialization of the GSDTO algorithm and compared algorithm, the number of agents is set to 10, and the number of iterations is set to 80. The distribution of agents is then set randomly for fair comparisons, as shown in Figure 4. The proposed GSDTO algorithm shows fast convergence than other algorithms. Figure 5 also depicts the average error versus objective function for the provided and compared algorithms. According to Figure 5, the proposed GSDTO algorithm shows better results than other algorithms based on average error.



Figure 4. Feature selection comparison of the presented and compared algorithms based on convergence curve.

The residual, QQ (quantile-quantile), homoscedasticity plots, and heat map of the presented and comparative binary methods are shown in Figure 6. As opposed to the plot of the original dataset, the potential issues can be seen in the residual values and plots. Some datasets are poor categorization candidates. A residual plot places the independent variable on the horizontal axis and the residual values on the vertical axis. If the residual values are evenly and randomly distributed across the horizontal axis, the ideal scenario is realized. When the mean and the total of the residuals are equal to zero, the residual value is determined as (Actual value-Predicted value). The residual plot is displayed in Figure 6. A residual plot's plot patterns can be used to identify whether a model is linear or nonlinear and which one is most appropriate. The homogeneity of variance or heteroscedasticity is visually inspected together with the projected scores for the dependent variable. A scenario known as homoscedasticity occurs when the error term, also known as noise or random disturbance in the relationship between the dependent and independent variables, is constant across all values of the independent variables. The accuracy of the research findings is increased by the heteroscedasticity plot, which is depicted in Figure 6. It can immediately and readily identify any violation.



Figure 5. The bGSDTO and compared algorithms' average error versus objective function.



Figure 6. Residual, QQ, homoscedasticity plots, and heat map of the bGSDTO and compared algorithms.

In Figure 6, the QQ plot is also displayed. An example is a probability plot. It is mostly used to compare two probability distributions by graphing the quantiles against one another. The point distributions in the QQ plot may be seen to roughly fit on the line in the figure. As a result, the relationship between the actual and anticipated residuals is linear, supporting the effectiveness of the suggested approach. The provided and compared algorithms' heat maps are displayed in Figure 6 as a data visualization tool. A two-dimensional color scale's intensity reveals an algorithm's level of sophistication. With regard to how the proposed method is superior to the comparable algorithms, the color variation provides clear visual indications. These figures confirm the quality of the bGSDTO algorithm in feature selection as mentioned in Figure 3.

4.2. Classification Scenario

The experiments' second scenario discusses the classification results of the presented GSDTO algorithm based on the LSTM classifier for 20 runs and 80 iterations using ten agents as mentioned in Table 3. The basic classifiers of NN, k-NN, and RF models and the proposed GSDTO algorithm-based LSTM method are applied to the selected features from the tested dataset in phase 1 as mentioned in Figure 3. The classification results of the proposed algorithm are compared with the WOA+LSTM, GWO+LSTM, GA+LSTM, and PSO+LSTM based models and the basic models to show the performance of the presented algorithm. There are four hyperparameters to train the LSTM model, which are fed to the proposed algorithm (GSDTO). These hyperparameters are the size of attention weights set N_a , encoding length for each attention weights L_e , size of champion attention weights subset W_a , and number of epochs T_e .

The performance metrics used in this part are the Mean Square Error (MSE) and the Area Under the ROC Curve (AUC). Table 9 presents the configuration parameters of basic classification models of NN, k-NN, and RF employed in this scenario. Table 10 shows the signal/basic classification model results of AUC and MSE for the basic classification models. The best results with the maximum AUC value of (0.797) and the minimum MSE value of (0.04887) are achieved by the RF model, followed by the NN model and then the k-NN model.

Classifier	Parameter (s)	Value (s)
NN	beta_1	0.6
	beta_2	0.899
	epsilon	$1 imes 10^{-6}$
	validation_fraction	0.1
	learning_rate_init	0.007
	hidden_layer_sizes	20
k-NN	leaf_size	20
	р	2
	n_neighbors	3
RF	min_weight_fraction_leaf	0.0
	min_samples_leaf	1
	n_estimators	40
	min_samples_split	2

Table 9. Parameters of basic classification models.

Table 10. Classification results of the three basic models for the tested dataset.

Model	AUC	MSE
NN	0.744	0.080226
K-NN	0.7387	0.098999
RF	0.797	0.04887

The classification results of the proposed and compared algorithms, based on optimizing the parameters of the LSTM model, are shown in Table 11. The results with the maximum AUC value of (0.9826) and the minimum MSE value of (0.00001413) are achieved by the GSDTO+LSTM model. The results show the model's superiority compared to the best basic classifier of the RF model, shown in Table 10, and the other models based on the LSTM method in Table 11. The WOA+LSTM-based model achieves the second best classification results with AUC of (0.957) and MSE of (0.000319), followed by the PSO+LSTM-based model, GWO+LSTM-based model, and the GA+LSTM-based model achieves the worst results.

Table 11. Proposed and compared algorithms classification results based on LSTM model.

	GSDTO+LSTM	WOA+LSTM	GWO+LSTM	GA+LSTM	PSO+LSTM
AUC	0.9826	0.957	0.943	0.934	0.949
MSE	0.00001413	0.000319	0.0002158	0.0007382	0.0004196

The presented GSDTO+LSTM and compared classifiers statistical descriptions and the Wilcoxon Signed-Rank test results are shown in Tables 12 and 13, respectively, based on 20 runs and 80 iterations for 10 agents as mentioned in Table 3, for fair comparison. This statistical test shows the significant difference between the results of the proposed algorithm (GSDTO+LSTM) and those of other algorithms with a *p*-value < 0.05.

	GSDTO+LSTM	WOA+LSTM	GWO+LSTM	GA+LSTM	PSO+LSTM
Number of values	20	20	20	20	20
Minimum	0.0000133	0.000124	0.000115	0.000641	0.000323
25% Percentile	0.0000133	0.000324	0.000215	0.000741	0.000423
Median	0.0000133	0.000324	0.000215	0.000741	0.000423
75% Percentile	0.0000133	0.000324	0.000215	0.000741	0.000423
Maximum	0.0000133	0.000424	0.0003315	0.0007854	0.000454
Range	0	0.0003	0.0002165	0.0001444	0.000131
10% Percentile	0.0000133	0.000324	0.000215	0.000741	0.000423
90% Percentile	0.0000133	0.000324	0.000215	0.000741	0.000423
Mean	0.0000133	0.000319	0.0002158	0.0007382	0.00042
Std. Deviation	0	0.00005104	0.00003521	$2.494 imes10^{-5}$	$2.38 imes10^{-5}$
Std. Error of Mean	0	0.00001141	$7.874 imes10^{-6}$	$5.576 imes10^{-6}$	$5.32 imes10^{-6}$
Coefficient of variation	0.000%	16.00%	16.32%	3.378%	5.666%
Geometric mean	0.0000133	0.000313	0.0002129	0.0007378	0.000419
Geometric SD factor	1	1.254	1.19	1.036	1.065
Harmonic mean	0.0000133	0.0003031	0.0002096	0.0007373	0.000418
Quadratic mean	0.0000133	0.0003229	0.0002185	0.0007386	0.00042
Skewness		-2.751	0.7003	-3.067	-3.734
Kurtosis		13.14	9.729	14.07	16.45
Sum	0.000266	0.00638	0.004317	0.01476	0.008391

Table 12. Proposed and compared classifiers description.

Table 13. Proposed and compared classifiers Wilcoxon Signed-Rank Test results.

	GSDTO+LSTM	WOA+LSTM	GWO+LSTM	GA+LSTM	PSO+LSTM
Theoretical median	0	0	0	0	0
Actual median	0.0000133	0.000324	0.000215	0.000741	0.000423
# values	20	20	20	20	20
Wilcoxon Signed-Rank					
Signed ranks' Sum (W)	210	210	210	210	210
Positive ranks' Sum	210	210	210	210	210
Negative ranks' Sum	0	0	0	0	0
P value (two tailed)	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001
Estimate or Exact?	Exact	Exact	Exact	Exact	Exact
Significant ($\alpha = 0.05$)?	Yes	Yes	Yes	Yes	Yes
How big is the discrepancy?					
Discrepancy	0.0000133	0.000324	0.000215	0.000741	0.000423

Figure 7 compares the presented and compared algorithms convergence curves based on the LSTM model. The proposed GSDTO+LSTM algorithm shows faster convergence than other algorithms. The box plot in Figure 8 indicates the MSE of the presented and compared algorithms-based LSTM model. This figure shows that the GSDTO+LSTM algorithm achieved minimum MSE results. Figure 9 shows the histogram of MSE for the presented and compared algorithms using the LSTM model, based on the number of values with the Bin Center range (0.0–0.00072), which confirms the stability of the proposed algorithm. Figures 8 and 9 present the visualization of changing the agents impact for the tested dataset for each run. Figure 10 shows the ROC curve for the presented GSDTO with PSO and WOA algorithms based LSTM model.



Figure 7. Comparison of the presented and compared algorithms convergence curves based LSTM model.



Figure 8. MSE for the presented and compared algorithms based LSTM model.



Figure 9. Histogram for the presented and compared algorithms based LSTM model.



Figure 10. ROC curve for the presented GSDTO with PSO and WOA algorithms based LSTM model.

The residual, QQ, homoscedasticity plots, and heat map of the presented and compared algorithms based on the LSTM model are shown in Figure 11. The residual plot places the independent variable on the horizontal axis and the residual values on the vertical axis. The residual plot is displayed in Figure 11. The homogeneity of variance or heteroscedasticity is visually inspected together with the projected scores for the dependent variable and can identify any violation. The accuracy of the research findings is increased by the heteroscedasticity plot, which is depicted in Figure 11. In Figure 11, the QQ plot is displayed, and it is used to compare two probability distributions by graphing the quantiles against one another. The point distributions in the QQ plot may be seen to roughly fit on the line in the figure. As a result, the relationship between the actual and anticipated residuals is almost linear, supporting the effectiveness of the suggested approach. The provided and compared algorithms' heat maps are displayed in Figure 11. A two-dimensional color scale's intensity reveals an algorithm's level of sophistication. These figures confirm the quality of the GSDTO+LSTM model as mentioned in Figure 3.



Figure 11. Residual, QQ, homoscedasticity plots, and heat map of the presented and compared algorithms based on LSTM model.

4.3. Validation and Discussion

The diagnostic accuracy of the suggested classification algorithm is compared to that of various DGA algorithms in the literature in Table 14. As shown in Table 14, the proposed classification algorithm has an overall diagnostic accuracy of 98.26 percent, which is higher than the diagnostic accuracy of other DGA techniques. The adaptive and NPR methods, which provide 94.6 and 90.54 percent accuracy, respectively, come close to the suggested algorithm. On the other hand, other classic DGA procedures, such as the Rogers Ratio method (45.95 percent), IEC 60599 (50 percent), and the Duval triangle method, have low overall diagnostic accuracies (66.27 percent). The diagnostic accuracy results for the suggested classification method indicated that it possesses a high capacity for correctly diagnosing transformer defects.

Table 14. The 74 testing samples diagnostic accuracy of the suggested algorithm compared to other techniques.

Fault Type	Samples	GSDTO+LSTM	Adaptive [2]	IEC-60599 [7]	IEC 60599 Modified [27]	Rog. Modified [27]
PD	7	100	100	28.57	100	100
D1	13	100	92.31	30.77	61.54	61.54
D2	24	95.83	91.67	41.67	87.5	79.17
T1	16	93.75	93.95	68.75	100	100
T2	4	100	100	75	100	100
T3	10	100	100	70	100	100
Overall	74	98.26	94.6	50	89.19	86.49
Fault Type	Samples	Duval [8,9]	NPR [53]	SVM [53]	Rog. 4 Ratios [6]	
PD	7	42.86	100	85.71	14.29	
D1	13	69.23	76.92	76.92	0	
D2	24	75	87.5	91.67	50	
T1	16	56.25	100	100	100	
T2	4	0	75	25	25	
T3	10	100	100	100	40	
Overall	74	66.27	90.54	87.84	45.95	

5. Sensitivity Analysis of the GSDTO Parameters

This section explores the GSDTO's parameter sensitivity analysis. The GSDTO consists of five parameters: the R-Parameter, the exploration percentage, the population size, the number of iterations, and the C-Parameter. These settings determine the performance of the algorithm when solving the evaluated optimization problem in this work. Any change to a parameter can influence the optimization technique. As a result, a sensitivity analysis of these parameters is conducted.

5.1. One-at-a-Time Sensitivity Analysis

The sensitivity analysis has been conducted using the One-at-a-Time (OAT) sensitivity measure [54]. OAT is recognized as one of the simplest strategies for sensitivity analysis. OAT examines the performance of an algorithm by varying a single parameter while leaving the others unchanged. Tables 15 and 16 detail the observed variations in GSDTO's time and fitness values as parameter values are adjusted. As shown in Tables 15 and 16, we selected 20 distinct values within the interval of each parameter by adding 5% to its length to generate a new evaluation value. Each of these variables underwent ten runs of the algorithm, and the averages for time and fitness are displayed in the table.

R-Param	eter	Explorati	on Percentage	Populati	on Size	Iterations	s Count	C-Param	eter
Values	Time	Values	Time	Values	Time	Values	Time	Values	Time
0.05	3.353	5	3.257	10	0.5	10	0.662	0.1	3.371
0.1	3.145	10	3.717	20	1.523	20	0.244	0.2	3.328
0.15	3.018	15	3.442	30	3.095	30	1.146	0.3	3.262
0.2	3.503	20	3.249	40	4.933	40	2.038	0.4	3.246
0.25	3.251	25	3.048	50	6.28	50	2.935	0.5	3.302
0.3	3.114	30	3.049	60	7.704	60	3.848	0.6	3.259
0.35	3.071	35	3.044	70	9.253	70	4.786	0.7	3.289
0.4	3.06	40	3.081	80	10.817	80	6.649	0.8	3.22
0.45	3.037	45	3.039	90	12.324	90	7.463	0.9	3.285
0.5	3.041	50	3.039	100	13.881	100	10.226	1	3.304
0.55	3.028	55	3.059	110	15.439	110	11.123	1.1	3.259
0.6	3.037	60	3.067	120	17.962	120	11.995	1.2	3.208
0.65	3.129	65	3.058	130	19.292	130	13.011	1.3	3.278
0.7	6.341	70	3.034	140	20.21	140	13.798	1.4	3.277
0.75	6.272	75	3.041	150	21.582	150	15.892	1.5	3.226
0.8	6.082	80	3.048	160	23.173	160	16.864	1.6	3.213
0.85	4.279	85	3.042	170	25.767	170	17.083	1.7	3.218
0.9	4.57	90	3.06	180	30.937	180	17.872	1.8	3.211
0.95	3.34	95	3.058	190	35.502	190	19.415	1.9	3.195
1	3.794	95	3.042	200	35.077	200	20.267	2	3.186

Table 15. Convergence time results for different values of the GSDTO's parameters.

 Table 16. Minimum Fitness results for different values of the GSDTO's parameters.

R-Parameter Exploration Percentage Pop		Populati	Population Size		Iterations Count		C-Parameter		
Values	Fitness	Values	Fitness	Values	Fitness	Values	Fitness	Values	Fitness
0.05	-11.2816	5	-9.1386	10	-9.6506	50	-7.5286	0.1	-8.0656
0.1	-11.2816	10	-9.1376	20	-12.3506	100	-6.9926	0.2	-8.0656
0.15	-11.8186	15	-11.2846	30	-11.2826	150	-8.6006	0.3	-6.9926
0.2	-11.8206	20	-10.7486	40	-11.2856	200	-8.6026	0.4	-8.0626
0.25	-11.2836	25	-11.2806	50	-10.2126	250	-8.0656	0.5	-8.0656
0.3	-11.2846	30	-10.7486	60	-12.3586	300	-8.0656	0.6	-6.9926
0.35	-10.7476	35	-10.7486	70	-12.3586	350	-9.6756	0.7	-8.0656
0.4	-11.2856	40	-11.2856	80	-12.3586	450	-8.0656	0.8	-8.0616
0.45	-10.7486	45	-11.2846	90	-11.8226	500	-9.1386	0.9	-6.9926
0.5	-11.2846	50	-10.2126	100	-12.3586	650	-10.2126	1	-8.0656
0.55	-11.2846	55	-11.2856	110	-12.3586	700	-8.6026	1.1	-8.0656
0.6	-12.3556	60	-11.8206	120	-12.3586	750	-10.2126	1.2	-8.0656
0.65	-11.2846	65	-11.2836	130	-12.3586	800	-10.7486	1.3	-9.1386
0.7	-10.7476	70	-11.2846	140	-12.3586	850	-9.1386	1.4	-10.2116
0.75	-10.7466	75	-11.2856	150	-12.3586	900	-9.6756	1.5	-11.2836
0.8	-12.3546	80	-10.7476	160	-12.3586	950	-10.7486	1.6	-9.1386
0.85	-11.2366	85	-12.3556	170	-12.3586	1000	-10.2126	1.7	-10.2116
0.9	-11.8096	90	-11.2716	180	-12.3586	1050	-10.7486	1.8	-12.3576
0.95	-12.3196	95	-11.2746	190	-12.3586	1150	-9.6756	1.9	-11.2846
1	-12.3306	95	-11.8096	200	-12.3586	1200	-10.2126	2	-12.3586

5.2. Regression Analysis

It has been determined through regression analysis how the algorithm's parameters can explain algorithm performance variations. Regression analysis is appropriate when we need to predict the value of a dependent variable (algorithm performance) based on the value of an independent variable (parameter). The results of the regression analysis for the GSDTO parameters, convergence time, and fitness are shown in Table 17. The value of R Square indicates the proportion of the total variance in time or fitness that can be explained by the parameter values. According to Table 17, the number of iterations and population size have the highest R Square value for convergence time, indicating that they can explain any variation in convergence time exceptionally well. In contrast, the exploration percentage and the R-Parameter can account for 60.3% and 78.5% of the variance in convergence time, respectively. In Table 17, significance F values less than 0.05 indicate that the regression model significantly predicts the algorithm's performance. The GSDTO's parameters convergence time and minimum fitness versus objective function are shown in Figure 12.

	Convergence	Time	Minimum Fitness		
Parameters	R Square	Significance F	R Square	Significance F	
R-Parameter	$7.85 imes 10^{-1}$	$1.44 imes 10^{-3}$	$8.85 imes 10^{-1}$	$9.16 imes10^{-4}$	
Exploration Percentage	$6.03 imes10^{-1}$	$1.67 imes10^{-5}$	$6.40 imes10^{-1}$	$3.91 imes 10^{-3}$	
Population Size	$8.06 imes10^{-1}$	$1.70 imes10^{-9}$	$9.80 imes10^{-1}$	$4.59 imes10^{-3}$	
Iterations Count	$8.23 imes10^{-1}$	$1.70 imes10^{-11}$	$7.44 imes10^{-1}$	$1.44 imes10^{-5}$	
C-Parameter	$4.06 imes10^{-1}$	$1.24 imes10^{-6}$	$9.57 imes10^{-1}$	$9.16 imes10^{-4}$	

Table 17. Results of regression analysis for the GSDTO's parameters.



Figure 12. The GSDTO's parameters convergence time and fitness versus objective function.

5.3. Statistical Significance

To determine whether there is a statistically significant difference between the means of the observations in Tables 15 and 16, we conducted an ANOVA on convergence time and fitness values while adjusting the GSDTO's parameters. The results of the GSDTO's ANOVA test are shown in Table 18. All p-values in Table 18 are less than 0.05. There is a statistically significant difference between the means of the five groups of convergence time and the five groups of minimal fitness observed by adjusting parameter values. A T-test with one tail was run at a significance level of 0.05. Table 19 provides the T-Test results for

each observed pair of convergence time and minimum fitness parameters for GSDTO. In the table, p-values less than 0.05 indicate a statistically significant difference between groups.

Table 18. ANOVA test results for the GSDTO's parameters of Convergence time and Fitness.

	SS	DF	MS	F (DFn, DFd)	p Value
Treatment (between columns)	11.54	9	1.282	F (9, 190) = 2483	<i>p</i> < 0.0001
Residual (within columns)	0.0981	190	0.0005163	-	-
Total	11.64	199	-	-	-

Table 19. T-test with one tail was run at a significance level of 0.05 for different values of the GSDTO's parameters.

	Convergence Time						
	R-Parameter	Exploration Percentage	Population Size	Iterations Count	C-Parameter		
Theoretical mean	0	0	0	0	0		
Actual mean	0.7775	0.593	0.8025	0.828	0.421		
Number of values	20	20	20	20	20		
One sample t test							
t, df	t = 93.03, df = 19	t = 48.00, df = 19	t = 72.62, df = 19	t = 93.97, df = 19	t = 38.47, df = 19		
<i>p</i> value (two tailed)	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001		
<i>p</i> value summary	****	****	****	****	****		
Significant (alpha = 0.05)?	Yes	Yes	Yes	Yes	Yes		
How big is the discrepancy?							
Discrepancy	0.7775	0.593	0.8025	0.828	0.421		
SD of discrepancy	0.03738	0.05525	0.04942	0.0394	0.04894		
SEM of discrepancy	0.008357	0.01235	0.01105	0.008811	0.01094		
95% confidence interval	0.7600 to 0.7949	0.5671 to 0.6189	0.7794 to 0.8256	0.8096 to 0.8464	0.3981 to 0.4439		
R squared (partial eta squared)	0.9978	0.9918	0.9964	0.9979	0.9873		
	Minimum Fitness						
	R-Parameter	Exploration Percentage	Population Size	Iterations Count	C-Parameter		
Theoretical mean	0	0	0	0	0		
Actual mean	0.885	0.6412	0.9764	0.759	0.937		
Number of values	20	20	20	20	20		
One sample t test							
t, df	t = 122.0, df = 19	t = 87.20, df = 19	t = 188.8, df = 19	t = 57.81, df = 19	t = 80.10, df = 19		
<i>p</i> value (two tailed)	< 0.0001	< 0.0001	< 0.0001	< 0.0001	< 0.0001		
<i>p</i> value summary	****	***	****	****	****		
Significant (alpha = 0.05)?	Yes	Yes	Yes	Yes	Yes		
How big is the discrepancy?							
Discrepancy	0.885	0.6412	0.9764	0.759	0.937		
SD of discrepancy	0.03244	0.03289	0.02313	0.05871	0.05231		
SEM of discrepancy	0.007255	0.007353	0.005172	0.01313	0.0117		
95% confidence interval	0.8698 to 0.9002	0.6258 to 0.6566	0.9656 to 0.9872	0.7315 to 0.7865	0.9125 to 0.9615		
R squared (partial eta squared)	0.9987	0.9975	0.9995	0.9943	0.997		

6. Conclusions and Future Work

This research proposed a novel meta-heuristic technique based on a dataset from real-technical systems to classify the dissolved gas analysis (DGA) for transformer faults diagnosis, which is one of the most vital elements in the electrical power system. Initially, the suggested binary (GSDTO) technique is employed to choose features from the evaluated dataset. The binary GSDTO (bGSDTO) method is evaluated against PSO, GWO, WOA, BBO, FA, GA, and BA. A classifier based on the proposed GSDTO algorithm and LSTM approach is then used on the tested dataset. Comparing the classification results with WOA+LSTM, GWO+LSTM, GA+LSTM, and PSO+LSTM. The GSDTO+LSTM algorithm's diagnostic accuracy is also examined using randomly selected data. Based on the statistical investigation, the robustness of the built model was examined. The results demonstrated that the developed model increased the diagnostic accuracy for all test cases to 98.26%. The sensitivity analysis of the GSDTO's parameters, R-Parameter, exploration percentage,

population size, number of iterations, and C-Parameter, confirm the performance of the algorithm. In the future, the binary GSDTO method and the GSDTO+LSTM-based classification algorithm can be generalized and evaluated on various datasets. Some additional experiments will be done to evaluate the scalability, runtime, and memory for the presented GSDTO+LSTM algorithm.

Author Contributions: Conceptualization, E.-S.M.E.-k., A.I., and A.A.A.; methodology, A.I.; software, E.-S.M.E.-k.; validation, S.S.M.G., N.B., and A.A.A.; formal analysis, S.S.M.G.; investigation, M.M.E.; resources, M.M.E.; data curation, S.S.M.G.; writing—original draft preparation, A.A.A. and A.I.; writing—review and editing, S.A.W., S.S.M.G., and N.B.; visualization, E.-S.M.E.-k.; supervision, S.A.W.; project administration, F.A.; funding acquisition, F.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Taif University Researchers Supporting Project TURSP-2020/97, Taif University, Taif, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors appreciate the Taif University Researchers Supporting Project TURSP-2020/97, Taif University, Taif, Saudi Arabia, for funding this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Badawi, M.; Ibrahim, S.A.; Mansour, D.E.A.; El-Faraskoury, A.A.; Ward, S.A.; Mahmoud, K.; Lehtonen, M.; Darwish, M.M.F. Reliable Estimation for Health Index of Transformer Oil Based on Novel Combined Predictive Maintenance Techniques. *IEEE Access* 2022, 10, 25954–25972, https://doi.org/10.1109/access.2022.3156102.
- Ghoneim, S.S.M.; Farrag, T.A.; Rashed, A.A.; El-Kenawy, E.S.M.; Ibrahim, A. Adaptive Dynamic Meta-Heuristics for Feature Selection and Classification in Diagnostic Accuracy of Transformer Faults. *IEEE Access* 2021, *9*, 78324–78340, https://doi.org/10.1109/access.2021.3083593.
- 3. Gouda, O.E.; El-Hoshy, S.H.; El-Tamaly, H.H. Proposed heptagon graph for DGA interpretation of oil transformers. *IET Gener. Transm. Distrib.* **2018**, *12*, 490–498, https://doi.org/10.1049/iet-gtd.2017.0826.
- Ward, S.A.; El-Faraskoury, A.; Badawi, M.; Ibrahim, S.A.; Mahmoud, K.; Lehtonen, M.; Darwish, M.M.F. Towards Precise Interpretation of Oil Transformers via Novel Combined Techniques Based on DGA and Partial Discharge Sensors. *Sensors* 2021, 21, 2223, https://doi.org/10.3390/s21062223.
- Benmahamed, Y.; Kherif, O.; Teguar, M.; Boubakeur, A.; Ghoneim, S.S.M. Accuracy Improvement of Transformer Faults Diagnostic Based on DGA Data Using SVM-BA Classifier. *Energies* 2021, 14, 2970, https://doi.org/10.3390/en14102970.
- Std C57.104-2008; IEEE Guide for the Interpretation of Gases Generated in Oil-Immersed Transformers. IEEE: Piscataway, NJ, USA, 2009; pp. 1–36, https://doi.org/10.1109/IEEESTD.2009.4776518.
- 7. IEC 60599: Mineral Oil-Filled Electrical Equipment in Service—Guidance on the Interpretation of Dissolved and Free Gases Analysis, Edition 2.1; IEC: Geneva, Switzerland, 2007.
- Duval, M. A review of faults detectable by gas-in-oil analysis in transformers. IEEE Electr. Insul. Mag. 2002, 18, 8–17, https://doi.org/10.1109/mei.2002.1014963.
- Duval, M.; dePabla, A. Interpretation of gas-in-oil analysis using new IEC publication 60599 and IEC TC 10 databases. *IEEE Electr. Insul. Mag.* 2001, 17, 31–41, https://doi.org/10.1109/57.917529.
- 10. Gouda, O.E.; El-Hoshy, S.H.; Ghoneim, S.S.M. Enhancing the Diagnostic Accuracy of DGA Techniques Based on IEC-TC10 and Related Databases. *IEEE Access* 2021, *9*, 118031–118041, https://doi.org/10.1109/access.2021.3107332.
- 11. Faiz, J.; Soleimani, M. Assessment of computational intelligence and conventional dissolved gas analysis methods for transformer fault diagnosis. *IEEE Trans. Dielectr. Electr. Insul.* 2018, 25, 1798–1806, https://doi.org/10.1109/tdei.2018.007191.
- 12. Hoballah, A.; Mansour, D.E.A.; Taha, I.B.M. Hybrid Grey Wolf Optimizer for Transformer Fault Diagnosis Using Dissolved Gases Considering Uncertainty in Measurements. *IEEE Access* 2020, *8*, 139176–139187, https://doi.org/10.1109/access.2020.3012633.
- 13. Li, X.; Wu, H.; Wu, D. DGA Interpretation Scheme Derived From Case Study. *IEEE Trans. Power Deliv.* 2011, 26, 1292–1293, https://doi.org/10.1109/tpwrd.2010.2091325.
- 14. The duval pentagon-a new complementary tool for the interpretation of dissolved gas analysis in transformers. *IEEE Electr. Insul. Mag.* **2014**, *30*, 9–12, https://doi.org/10.1109/mei.2014.6943428.
- 15. Cheim, L.; Duval, M.; Haider, S. Combined Duval Pentagons: A Simplified Approach. *Energies* 2020, 13, 2859, https://doi.org/10.3390/en13112859.

- 16. Mansour, D.E.A. Development of a new graphical technique for dissolved gas analysis in power transformers based on the five combustible gases. *IEEE Trans. Dielectr. Electr. Insul.* **2015**, *22*, 2507–2512, https://doi.org/10.1109/tdei.2015.004999.
- Miranda, V.; Castro, A. Improving the IEC Table for Transformer Failure Diagnosis With Knowledge Extraction From Neural Networks. *IEEE Trans. Power Deliv.* 2005, 20, 2509–2516, https://doi.org/10.1109/tpwrd.2005.855423.
- Souahlia, S.; Bacha, K.; Chaari, A. MLP neural network-based decision for power transformers fault diagnosis using an improved combination of Rogers and Doernenburg ratios DGA. *Int. J. Electr. Power Energy Syst.* 2012, 43, 1346–1353, https://doi.org/10.1016/j.ijepes.2012.05.067.
- 19. Guardado, J.; Naredo, J.; Moreno, P.; Fuerte, C. A comparative study of neural network efficiency in power transformers diagnosis using dissolved gas analysis. *IEEE Trans. Power Deliv.* 2001, *16*, 643–647, https://doi.org/10.1109/61.956751.
- 20. Equbal, M.D.; Khan, S.A.; Islam, T. Transformer incipient fault diagnosis on the basis of energy-weighted DGA using an artificial neural network. *Turk. J. Electr. Eng. Comput. Sci.* 2018, 26, 77–88.
- Ou, M.; Wei, H.; Zhang, Y.; Tan, J. A Dynamic Adam Based Deep Neural Network for Fault Diagnosis of Oil-Immersed Power Transformers. *Energies* 2019, 12, 995, https://doi.org/10.3390/en12060995.
- Abu-Siada, A.; Hmood, S.; Islam, S. A new fuzzy logic approach for consistent interpretation of dissolved gas-in-oil analysis. IEEE Trans. Dielectr. Electr. Insul. 2013, 20, 2343–2349, https://doi.org/10.1109/tdei.2013.6678888.
- 23. Noori, M.; Effatnejad, R.; Hajihosseini, P. Using dissolved gas analysis results to detect and isolate the internal faults of power transformers by applying a fuzzy logic method. *IET Gener. Transm. Distrib.* 2017, *11*, 2721–2729, https://doi.org/10.1049/iet-gtd.2017.0028.
- Mulyodinoto, K.U.; Suwarno.; Prasojo, R.; Abu-Siada, A. Applications of ANFIS to Estimate the Degree of Polymerization Using Transformer Dissolve Gas Analysis and Oil Characteristics. *Polym. Sci.* 2018, 14, 1–9.
- 25. Bacha, K.; Souahlia, S.; Gossa, M. Power transformer fault diagnosis based on dissolved gas analysis by support vector machine. *Electr. Power Syst. Res.* **2012**, *83*, 73–79, https://doi.org/10.1016/j.epsr.2011.09.012.
- Zhang, Y.; Li, X.; Zheng, H.; Yao, H.; Liu, J.; Zhang, C.; Peng, H.; Jiao, J. A Fault Diagnosis Model of Power Transformers Based on Dissolved Gas Analysis Features Selection and Improved Krill Herd Algorithm Optimized Support Vector Machine. *IEEE Access* 2019, 7, 102803–102811, https://doi.org/10.1109/access.2019.2927018.
- Taha, I.B.M.; Hoballah, A.; Ghoneim, S.S.M. Optimal ratio limits of rogers' four-ratios and IEC 60599 code methods using particle swarm optimization fuzzy-logic approach. *IEEE Trans. Dielectr. Electr. Insul.* 2020, 27, 222–230, https://doi.org/10.1109/tdei.2019.008395.
- Illias, H.A.; Chai, X.R.; Bakar, A.H.A.; Mokhlis, H. Transformer Incipient Fault Prediction Using Combined Artificial Neural Network and Various Particle Swarm Optimisation Techniques. PLoS ONE 2015, 10, e0129363, https://doi.org/10.1371/journal.pone.0129363.
- Illias, H.A.; Chai, X.R.; Bakar, A.H.A. Hybrid modified evolutionary particle swarm optimisation-time varying acceleration coefficient-artificial neural network for power transformer fault diagnosis. *Measurement* 2016, 90, 94–102, https://doi.org/10.1016/j.measurement.2016.04.052.
- Ghoneim, S.S.M.; Mahmoud, K.; Lehtonen, M.; Darwish, M.M.F. Enhancing Diagnostic Accuracy of Transformer Faults Using Teaching-Learning-Based Optimization. *IEEE Access* 2021, 9, 30817–30832, https://doi.org/10.1109/access.2021.3060288.
- Bello, R.; Gomez, Y.; Nowe, A.; Garcia, M.M. Two-Step Particle Swarm Optimization to Solve the Feature Selection Problem. In Proceedings of the Seventh International Conference on Intelligent Systems Design and Applications (ISDA 2007), Rio de Janeiro, Brazil, 20–24 October 2007; pp. 691–696, https://doi.org/10.1109/ISDA.2007.101.
- El-Kenawy, E.S.M.; Eid, M.M.; Saber, M.; Ibrahim, A. MbGWO-SFS: Modified Binary Grey Wolf Optimizer Based on Stochastic Fractal Search for Feature Selection. *IEEE Access* 2020, *8*, 107635–107649, https://doi.org/10.1109/access.2020.3001151.
- Eid, M.M.; El-kenawy, E.S.M.; Ibrahim, A. A binary Sine Cosine-Modified Whale Optimization Algorithm for Feature Selection. In Proceedings of the 2021 National Computing Colleges Conference (NCCC), Taif, Saudi Arabia, 27–28 March 2021; https://doi.org/10.1109/nccc49330.2021.9428794.
- 34. Simon, D. Biogeography-Based Optimization. IEEE Trans. Evol. Comput. 2008, 12, 702-713, https://doi.org/10.1109/tevc.2008.919004.
- 35. Fister, I.; Yang, X.S.; Fister, I.; Brest, J. Memetic Firefly Algorithm for Combinatorial Optimization. *arXiv* 2012, arXiv:1204.5165.
- Kabir, M.M.; Shahjahan, M.; Murase, K. A new local search based hybrid genetic algorithm for feature selection. *Neurocomputing* 2011, 74, 2914–2928, https://doi.org/10.1016/j.neucom.2011.03.034.
- Karakonstantis, I.; Vlachos, A. Bat algorithm applied to continuous constrained optimization problems. J. Inf. Optim. Sci. 2021, 42, 57–75, https://doi.org/10.1080/02522667.2019.1694740.
- Egyptian Electricity Holding Company (EEHC) Reports. Available online: http://www.moee.gov.eg/english_new/report.aspx (accessed on 18 April 2022).
- Agrawal, S.; Chandel, A.K. Transformer incipient fault diagnosis based on probabilistic neural network. In Proceedings of the 2012 Students Conference on Engineering and Systems, Allahabad, India, 16–18 March 2012; IEEE: Piscataway, NJ, USA, 2012; https://doi.org/10.1109/sces.2012.6199110.
- 40. Wang, M.H. A novel extension method for transformer fault diagnosis. *IEEE Trans. Power Deliv.* 2003, 18, 164–169, https://doi.org/10.1109/tpwrd.2002.803838.
- Zhu, Y.; Wang, F.; Geng, L. Transformer Fault Diagnosis Based on Naive Bayesian Classifier and SVR. In Proceedings of the TENCON 2006–2006 IEEE Region 10 Conference, Hong Kong, China, 14–17 November 2006; IEEE: Piscataway, NJ, USA, 2006; https://doi.org/10.1109/tencon.2006.343895.

- Siva Sarma, D.V.S.S.; Kalyani, G.N.S. ANN approach for condition monitoring of power transformers using DGA. In Proceedings of the 2004 IEEE Region 10 Conference TENCON 2004, Chiang Mai, Thailand, 24 November 2004; Volume 3, pp. 444–447, https://doi.org/10.1109/TENCON.2004.1414803.
- Hu, J.; Zhou, L.; Song, M. Transformer Fault Diagnosis Method of Gas Hromatographic Analysis Using Computer Image Analysis. In Proceedings of the 2012 Second International Conference on Intelligent System Design and Engineering Application, Sanya, China, 6–7 January 2012; pp. 1169–1172, https://doi.org/10.1109/ISdea.2012.599.
- Seifeddine, S.; Khmais, B.; Abdelkader, C. Power transformer fault diagnosis based on dissolved gas analysis by artificial neural network. In Proceedings of the 2012 First International Conference on Renewable Energies and Vehicular Technology, Nabeul, Tunisia, 26–28 March 2012; pp. 230–236, https://doi.org/10.1109/REVET.2012.6195276.
- Rajabimendi, M.; Dadios, E.P. A hybrid algorithm based on neural-fuzzy system for interpretation of dissolved gas analysis in power transformers. In Proceedings of the TENCON 2012 IEEE Region 10 Conference, Cebu, Philippines, 19–22 November 2012; pp. 1–6, https://doi.org/10.1109/TENCON.2012.6412171.
- Zhang, G.; Yasuoka, K.; Ishii, S.; Li Yang.; Zhang Yan. Application of fuzzy equivalent matrix for fault diagnosis of oil-immersed insulation. In Proceedings of the 1999 IEEE 13th International Conference on Dielectric Liquids (ICDL'99) (Cat. No. 99CH36213), Nara, Japan, 25 July 1999; pp. 400–403, https://doi.org/10.1109/ICDL.1999.798957.
- Gouda, O.E.; Saleh, S.M.; El-hoshy, S.H. Power Transformer Incipient Faults Diagnosis Based on Dissolved Gas Analysis. *Indones. J. Electr. Eng. Comput. Sci.* 2016, 17, 10–16, https://doi.org/10.11591/ijeecs.v1.i1.pp10-16.
- El-Kenawy, E.S.M.; Mirjalili, S.; Ibrahim, A.; Alrahmawy, M.; El-Said, M.; Zaki, R.M.; Eid, M.M. Advanced Meta-Heuristics, Convolutional Neural Networks, and Feature Selectors for Efficient COVID-19 X-Ray Chest Image Classification. *IEEE Access* 2021, 9, 36019–36037, https://doi.org/10.1109/access.2021.3061058.
- Ibrahim, A.; Mirjalili, S.; El-Said, M.; Ghoneim, S.S.M.; Al-Harthi, M.M.; Ibrahim, T.F.; El-Kenawy, E.S.M. Wind Speed Ensemble Forecasting Based on Deep Learning Using Adaptive Dynamic Optimization Algorithm. *IEEE Access* 2021, *9*, 125787–125804, https://doi.org/10.1109/access.2021.3111408.
- Takieldeen, A.E.; El-kenawy, E.S.M.; Hadwan, M.; Zaki, R.M. Dipper Throated Optimization Algorithm for Unconstrained Function and Feature Selection. *Comput. Mater. Contin.* 2022, 72, 1465–1481, https://doi.org/10.32604/cmc.2022.026026.
- El-kenawy, E.S.M.; Mirjalili, S.; Ghoneim, S.S.M.; Eid, M.M.; El-Said, M.; Khan, Z.S.; Ibrahim, A. Advanced Ensemble Model for Solar Radiation Forecasting using Sine Cosine Algorithm and Newton's Laws. *IEEE Access* 2021, 9, 115750–115765, https://doi.org/10.1109/access.2021.3106233.
- 52. Emary, E.; Zawbaa, H.M.; Hassanien, A.E. Binary grey wolf optimization approaches for feature selection. *Neurocomputing* **2016**, 172, 371–381, https://doi.org/10.1016/j.neucom.2015.06.083.
- Taha, I.B.; Dessouky, S.S.; Ghoneim, S.S. Transformer fault types and severity class prediction based on neural pattern-recognition techniques. *Electr. Power Syst. Res.* 2021, 191, 106899, https://doi.org/10.1016/j.epsr.2020.106899.
- Confalonieri, R.; Bellocchi, G.; Bregaglio, S.; Donatelli, M.; Acutis, M. Comparison of sensitivity analysis techniques: A case study with the rice model WARM. *Ecol. Model.* 2010, 221, 1897–1906, doi:https://doi.org/10.1016/j.ecolmodel.2010.04.021.