IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 June 2022, accepted 1 August 2022, date of publication 5 August 2022, date of current version 15 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3196660

RESEARCH ARTICLE

Pothole and Plain Road Classification Using Adaptive Mutation Dipper Throated Optimization and Transfer Learning for Self Driving Cars

AMEL ALI ALHUSSAN[1], DOAA SAMI KHAFAGA[1], EL-SAYED M. EL-KENAWY[2], (Senior Member, IEEE), ABDELHAMEED IBRAHIM[3], (Member, IEEE), MARWA METWALLY EID[4], (Member, IEEE), AND ABDELAZIZ A. ABDELHAMID[5,6]

[1]Department of Computer Sciences, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia
[2]Department of Communications and Electronics, Delta Higher Institute of Engineering and Technology (DHIET), Mansoura 35111, Egypt
[3]Computer Engineering and Control Systems Department, Faculty of Engineering, Mansoura University, Mansoura 35516, Egypt
[4]Faculty of Artificial Intelligence, Delta University for Science and Technology, Mansoura 35712, Egypt
[5]Department of Computer Science, Faculty of Computer and Information Sciences, Ain Shams University, Cairo 11566, Egypt
[6]Department of Computer Science, College of Computing and Information Technology, Shaqra University, Shaqra 11961, Saudi Arabia

Corresponding authors: Doaa Sami Khafaga (dskhafga@pnu.edu.sa) and El-Sayed M. El-Kenawy (skenawy@ieee.org)

This work was supported in part by the Princess Nourah bint Abdulrahman University Researchers Supporting Project under Grant PNURSP2022R308; and in part by Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

ABSTRACT Self-driving car plays a crucial role in implementing traffic intelligence. Road smoothness in front of self-driving cars has a significant impact on the car's driving safety and comfort. Having potholes on the road may lead to several problems, including car damage and the occurrence of collisions. Therefore, self-driving cars should be able to change their driving behavior based on the real-time detection of road potholes. Various methods are followed to address this problem, including reporting to authorities, employing vibration-based sensors, and 3D laser imaging. However, limitations, such as expensive setup costs and the danger of discovery, affected these methods. Therefore, it is necessary to automate the process of potholes identification with sufficient precision and speed. A novel method based on adaptive mutation and dipper throated optimization (AMDTO) for feature selection and optimization of the random forest (RF) classifier is presented in this paper. In addition, we propose a new adaptive method for dataset balancing, referred to as optimized hashing SMOTE, to boost the performance of the optimized model. Data on potholes in different weather conditions and circumstances were collected and augmented before training the proposed model. The effectiveness of the proposed method is shown in experiments in classifying road potholes accurately. Eleven feature selection methods, including WOA, GWO, and PSO, and three machine learning classifiers were included in the conducted experiments to measure the superiority of the proposed method. The proposed method, AMDTO+RF, achieved a pothole classification accuracy of (99.795%), which outperforms the accuracy achieved by the other approaches, WOA+RF of 97.5%, GWO+RF of 98.6%, PSO+RF of 98.1%, and transfer learning approaches, AlexNet of 86.8%, VGG-19 of 87.3%, GoogLeNet of 90.4%, and ResNet-50 of 93.8%. In addition, an in-depth statistical analysis is performed on the recorded results to study the significance and stability of the proposed method.

INDEX TERMS Potholes classification, dipper throated optimization, particle swarm optimization, adaptive mutation, optimized SMOTE, feature selection, random forest.

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

I. INTRODUCTION

One of the most essential solutions for implementing traffic intelligence in intelligent transportation is self-driving cars. Scientific research organizations and vehicle manufacturers have recently been paying close attention to autonomous driving. It's one of the most promising scientific and technical research areas, as well as a potential solution for improving driving safety, decreasing traffic congestion, and reducing air pollution [1].

Self-driving cars must have a high level of safety. High-level intelligent behavior, such as obstacle avoidance and path planning for self-driving cars, is predicated on environmental awareness. Detecting and identifying road potholes is one of the most crucial activities in environmental perception. The capture of road data in real-time while driving can give critical information for real-time course planning, which can help assure vehicle safety [2].

For self-driving cars, accurate road recognition is challenging. GPS, sensors, and high-resolution digital map guidance are all used in the autonomous driving process. A self-driving car can detect its direction by merging environmental data collected by digital cameras and light detection and ranging (LiDAR) sensors around the vehicle with GPS coordinates on a high-definition digital map to evaluate the current road conditions [3].

Various cameras and sensors have been installed on self-driving cars to collect road information in order to gather rapid, accurate, and trustworthy road information for decision-making. The vehicle vision system is the first system of interest in self-driving cars. Image acquisition devices collect images, which are subsequently transformed into digital signals based on pixel distribution, brightness, and color. Finally, by extracting the features of the digital signals, the needed information is identified. The second is LiDAR, which can gather point cloud data constantly while driving to analyze the surroundings around the vehicle. A LiDAR's laser beams scan the surrounding surroundings in a repeated manner, producing road point cloud data. Environmental sensing also utilizes ultrasonic and infrared sensors [4].

In principle, the most crucial job for self-driving cars is to determine the drivable region and avoid the garbled surface. However, because the self-driving vehicle's planned control algorithm requires greater lane change space than a typical driver for safety reasons, the self-driving vehicle may strive to avoid lane changes in a short distance in practice. When self-driving cars are in front of and surrounding them, a follow-up order is likely to be made. The flatness of the road ahead of the self-driving vehicle becomes a significant element in limiting the self-driving vehicle's safety, and stability [5].

The high-precision map, in combination with LiDAR and other sensors, allows self-driving cars to perceive their surroundings. However, because the high-precision map only gets updated once a week, self-driving cars have to get road information through real-time sensing if there are new potholes or bumps on the road. Furthermore, when there is no high-precision map on the unusual road, real-time road recognition becomes much more crucial. The radar detection findings will be influenced if the pothole road is used by other cars, which is not beneficial to automated driving safety. As a result, to improve the safety of self-driving cars, it is critical to detect road information in real-time [6].

A significant number of road accidents and automotive damage is caused by road potholes. Due to the increase in vehicle traffic and pollution, potholes of all sizes have lately developed on highways. In most countries, the states and municipal governments usually assign a significant amount of funds to fix potholes; however, major injuries or deaths still occur as a consequence of the spread of these potholes. Several methods have been employed to detect potholes. Vibration detection and 3D reconstruction methods are only a few of the options available. Road potholes may be automatically identified using a variety of technologies, including sensor systems and 3D reconstruction techniques (laser and stereo vision-based), image processing techniques, and model building (machine-learning techniques and deep learning techniques) [7].

Senor-based approaches employ sensors that sense vibration to discover potholes. Detecting potholes may be hampered by the vibration sensor's inability to distinguish between potholes and road joints. In other order to locate potholes on the road's surface, 3D reconstruction methods collect 3D road data. Setup and configuration cost a lot of money, and there's a chance that the cameras may be out of alignment, reducing their accuracy in detecting things on the ground. To identify potholes, traditional image processing algorithms need tedious processes, for instance, manually extracting features and changing the parameters of image processing and stages to account for the various conditions of roads. Model-based pothole detection techniques have been developed thanks to advances in image processing and low-cost camera equipment. Traditional algorithms of machine learning were utilized to locate potholes in 2D digital pictures. A lot of processing power allows them to get exact results [8]. Figure 1 shows the pothole detection process by the self-driving car. Table 1 presents the challenges of the existing pothole detection methods.

Machine learning (ML) techniques for identifying potholes need experts to manually extract attributes in order to improve their accuracy. For deep learning, deep convolutional neural networks (CNNs) are used, which can extract and classify information simultaneously [21], [22]. Single Shot Multibox (SSD) and You Only Look Once (YOLO) detectors are two examples of one-stage pothole detectors that have been created and published. The primary benefits of employing this technology are its moderate accuracy and speedy detecting speed. Faster R-CNN, a two-stage pothole detector, has gotten little attention in the scientific community. They can achieve high accuracy while still moving at a tolerable pace because of their moderate detecting speed [23].

TABLE 1. Challenges of the existing pothole detection methods.

Approach	Challenges
Sensor-based [9, 10, 11]	* It needs special devices * Road conditions may cause damage to sensors * The center of the lane results in false-negative potholes * Road-joints are classified as potholes * Information about the shape and area of potholes is lacking
3D Reconstruction [12, 13]	* The 3D laser scanner is expensive * The reconstruction of a surface is computationally expensive * It needs repetitive camera alignments
Image processing [14, 15]	* Several parameters need to be adjusted * Complexity of this approach is high * Hard to be realized in real-time
Model-based [16, 17]	* Large dataset is needed * Feature extraction is usually performed manually * The developed models are shallow
Deep learning [18, 19, 20]	* Large dataset is needed * A balance between detection time and model accuracy is required

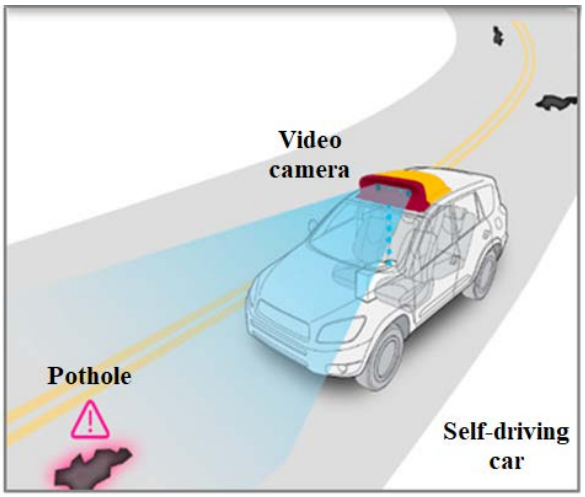


FIGURE 1. Pothole detection by self-driving car.

Feature engineering is a vital process for all machine learning techniques. This process entails the selection and extraction of the relevant features that are necessary for machine learning pipelines [24], [25]. In the literature, the words feature selection and feature extraction are used interchangeably. However, there is a distinction between them in essence. The process of feature extraction focuses on processing the raw data to extract additional variables that can help the algorithms of machine learning to work properly [26], [27]. On the other hand, the process of feature selection is interested in selecting and identifying the relevant features from the dataset that fulfill certain conditions, such as uniqueness, consistency, and meaningfulness. To realize the feature selection process, binary values (0 and 1) are used to constrain the search space. Consequently, an update is required to be applied to the continuous values-based optimizers to allow them to work properly with this issue.

The feature selection is the most significant step in feature engineering as it selects the most appropriate features that enable optimizers to achieve the best performance [28], [29]. The feature selection task can be defined as a binary vector of n-features, with each feature having a value of 1 or 0, depending on whether it is included in the solution or not. A random population of vectors with random features is usually the starting point for the meta-heuristic algorithms that are followed by a series of explorations and exploitation to find the optimum collection of features [30], [31].

Because self-driving cars must make quick decisions, identifying potholes is important to ensuring driving safety. To the best of the authors' knowledge, however, few research efforts have focused on detecting and identifying potholes in self-driving cars. Table 2 shows the main research approaches for pothole detection in self-driving cars. The method in [1] can only be suitable for real-time systems. Authors in [8] presented a method that fitted to be used under an edge computing schema. Models deployed in [32] should be run based on a cloud-based web server. The method in [33] achieved 100% accuracy, but it is limited by the sample size used in this work. Algorithms deployed in [34] had the issue of limited hardware/software devices.

Therefore, in this paper, a new method for self-driving cars to accurately detect road potholes to increase driving safety is proposed. The proposed method employs a deep convolutional deep neural network that works as a feature extractor. The extracted features are fed to the random forest classifier to give the final decision. To boost the performance of the random forest, an advanced mutation and dipper throated optimization (AMDTO) is proposed to optimize the parameter of the classifier. Several experiments were conducted to prove the proposed method's effectiveness. In this research, we propose three algorithms based on meta-heuristic optimization for detecting road potholes accurately. The main contributions of this work are listed in the following:

TABLE 2. Main research approaches for pothole detection.

Paper	Target	Methodology	Data Acquisition	Results
[1]	Classify the road general condition: "normal road" and "pothole"	Train various classification models of machine learning.	Accelerometer and GPS data obtained through a purpose-built mobile application.	Precision and recall rate exceeds 95%
[8]	Detect potholes, speed bumps, and metal bumps.	QF-COTE. Threshold detection and sliding window algorithm.	Legacy datasets, simulated data through Carsim.	Method fitted to be used under an edge computing schema.
[32]	Classify "pothole" and "non-pothole", good or bad road conditions.	Use of SVM model to classify good-bad roads and predict potholes.	Gyroscope and accelerometer data, speed, and GPS location of the vehicle, using two iOS applications.	Road condition: 93.4% accuracy, Pothole detection: 78% accuracy and 42% recall
[35]	Pothole occurrences	Use of a Neural network based on ReLU activation function.	Accelerometer and gyroscope data from smartphone mounted on the windshield.	94% of accuracy and 81% of recall were reported. Suitable for real-time systems.
[33]	Pothole detection	Implementation of an algorithm based on dynamic threshold detection, using three-axis accelerometer data.	Mobile sensing, through accelerometer data normalization.	100% accuracy.
[34]	Large and small potholes, clusters of potholes, gaps, drain pits.	Set of algorithms based on accelerometer data and threshold definition: Z-THRESH, Z-DIFF, STDEV(Z), G-ZERO.	Preliminary data gathered from a modified LynxNet collar device.	Algorithms deployed on limited hardware/software devices. True positive rates as high as 90%.
[36]	Potholes, maintenance holes, transverse cracks, longitudinal cracks, railroad tracks, speed-bumps, deceleration strips, paved roads, and road dents.	The obtained signals were de-noised using wavelets, and data sets are "time windowed", The algorithm applied feature extraction techniques.	Multiple IMUs, GPS receivers, smart devices, low-cost MEMS, mounted on a testbed.	Multi-level SVM classifier, average TPR performance of 90%
[37]	Detect street anomalies using sensors within the vehicles that travel daily and connecting them to a fog-computing architecture on a V2I network.	Use of neural Network and a K-Nearest Neighbor to select the best option to handle the acquired data.	An instrumented vehicle obtained the reference through accelerometry sensors and sent the data through a mid-range communication system.	Both algorithms had over 90% accuracy during training, with the KNN having 95.55% accuracy and the ANN scoring 96.79% over the training data.

- A new preprocessing approach for imbalanced the road potholes dataset based on optimized hash SMOTE (Synthetic Minority Oversampling Technique) is proposed.
- A new feature selection algorithm based on binary advanced mutation for selecting the best set of features based on the road potholes dataset can boost the classification accuracy.
- A new hybrid optimization algorithm based on advanced mutation and dipper throated optimization (AMDTO) for learning the parameters of a random forest model.
- A one-sample two-tailed t-test and ANOVA test are employed to assess the statistical significance of the proposed optimization algorithms.
- Statistical analysis is performed to evaluate the performance of the proposed approach for pothole and plain road classification.
- A complexity analysis of the proposed algorithm is provided.
- Robust classification of road potholes.
- The generalization capability of the proposed algorithm to other datasets.

The organization of this paper goes as follows. The literature review is presented in section 2. The proposed methodology is presented and explained in section 3. Then the achieved results are discussed in section 4. Finally, section 5 concludes the findings of this research.

II. LITERATURE REVIEW

The detection of potholes may be accomplished by using one or more of the following four approaches. 1) sensors, 2) 3D reconstruction, 3) image processing, and 4) deep learning-based models. In this section, the recent research attempts based on these approaches are presented and discussed. These methods are based on the embedded sensors installed in the self-driving cars shown in Fig 2.

A. POTHOLE DETECTION USING SENSORS

Potholes are being found, and the integrity of the pavement surface is being studied using a variety of vibration sensors (such as an oscilloscope or accelerometer or on bikes, cars, and buses). Vibration sensors on a computer may be built-in or installed outside, depending on the application. Using a machine-learning method, authors in [16] discovered potholes and other major surface imperfections using GPS sensors and three-axis accelerometers. They looked at xz-ratio (xz-peak), z-peak, speed, and high-pass vs. z-ratio (xz-speed). These considerations also ruled out incidents like walking on railroad wires or beating on a door.

Authors in [38] utilized backward and forwarded selection, support vector machine, and genetic algorithm with principal component analysis in an attempt to minimize the number of attributes. On devices with limited technology, sensor-based pothole detection techniques are challenging to implement since potholes in the middle of the lane are impossible to detect because they aren't impacted by any car wheels [18]. Potholes in the middle of the lane aren't touched by any vehicle wheels; hence the sensors may provide positive or negative results that are incorrect.

B. POTHOLE DETECTION USING 3D RECONSTRUCTION

There are two types of 3D reconstruction techniques: laser-based and stereo-vision-based. A 3D laser scanner uses reflected laser pulses to build exact digital reconstructions of objects. With these lasers, the depth of potholes can be measured in real-time. To identify potholes, authors in [39] developed a system that employs a light source to cast a pattern of laser beams onto the pavement, a camera to capture the pavement lit by the laser beams, and image processing on the collected photographs. Procedures including Multi-window Median filtering, Tile Partitions with shared thresholding, Laser line deformation, and Template matching were examined. Potholes may be seen promptly using laser-based technologies. Installing a 3D laser scanner in an automobile isn't

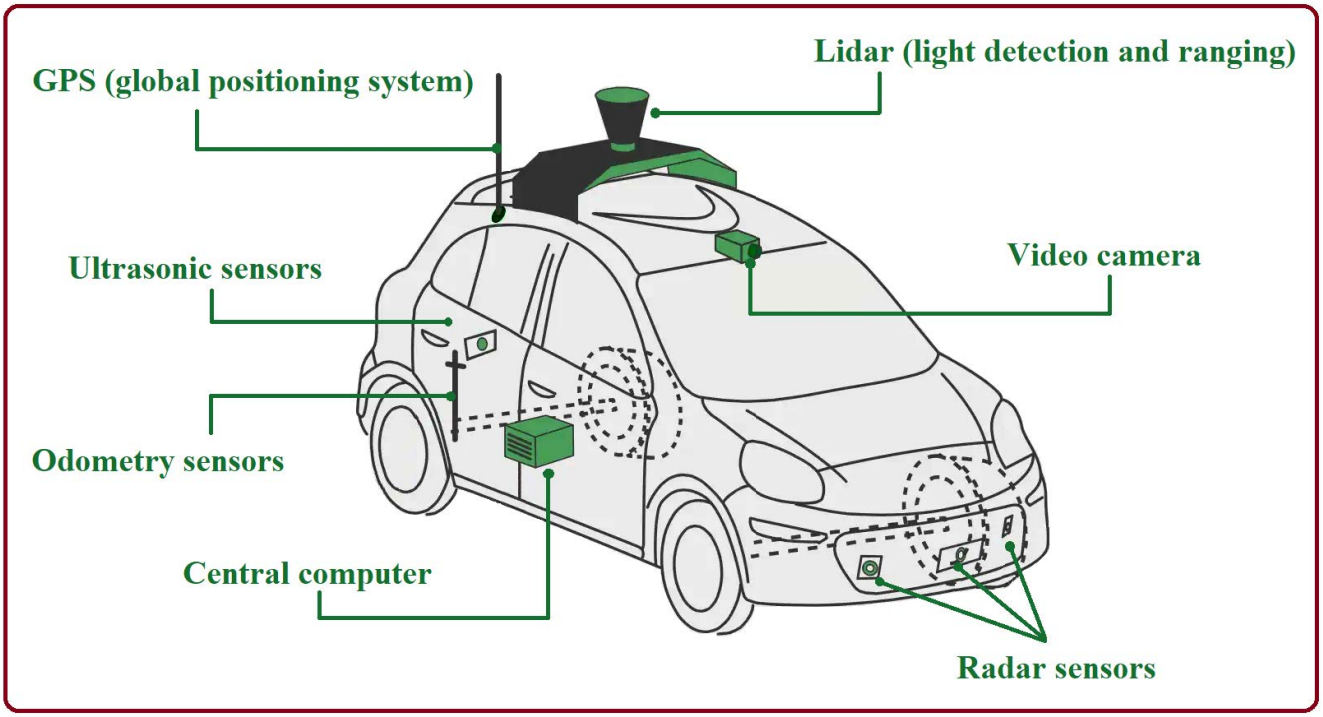


FIGURE 2. A Self-driving car with embedded sensors and central computer.

cheap, to be sure. 3D information can be extracted from digital photographs using stereo vision techniques. Potholes may be detected using stereo vision techniques, which are being applied in various research projects.

Authors in [40] captured images of potholes from both sides using a stereo camera. The generation of a disparity map was accomplished by the use of a computationally efficient strategy. The road surface and potholes can be estimated using a low-computing bi-square weighted resilient least-squares strategy. Geometric coordinates for potholes may be used to access pothole properties, such as volume and size, to aid in prioritizing repairs. Setup costs are a typical issue with laser and stereo-vision technologies. (1) Stereo-vision systems are inefficient due to the large amount of computation necessary to reconstruct the pavement surface. The movement of the vehicle and the misalignment of the camera might impair the quality of the picture.

C. POTHOLE DETECTION USING IMAGE PROCESSING

Object detectors used in image processing rely on low-level features about objects that are extracted using hand-crafted representations. Authors in [36] employed edge detection and morphological processing to identify contours, which were subsequently processed using the Hough transform algorithm to extract information. In order to create blurry grayscale images, they combined many frames. Authors in [41] claim that potholes in an asphalt pavement may be identified using fuzzy C-means clustering and morphological reconstruction methods. Additionally, authors in [38] used image processing to find potholes on highways and eliminate irrelevant objects, such as automobiles and plants. Canny filters and contour detection are used to find potholes in photos. For the studies, the accuracy was found to be 81.8%, while the recall rate was found to be 74.44%.

Testing on various kinds of roads may not provide results as accurate as those in the images, despite their high accuracy ratings in those photographs. This is how the authors in [42] find potholes: Pre-processing to remove dark areas from a grayscale image, candidate extraction to find the vanishing point to build virtual lanes, and cascade detection to extract the pothole region using threshold values. With a 71% recall rate, an 88% success rate was achieved. Stages 1 through 3 are used to identify potholes. By comparing the properties of the target and the backdrop, one may determine if a candidate area is a pothole or avoid it by comparing its size and compactness to those of the target. Using image processing methods to detect potholes may be inaccurate if there is a pothole on the road or the pothole size changes in size. To account for different road conditions, these approaches need time-consuming modifications to the image processing settings and procedures. These methods can't be used to identify potholes in the road in real-time because of their processing complexity.

D. POTHOLES DETECTION USING MODEL-BASED APPROACHES

In recent years, machine learning (ML) techniques have been more popular in the development of trained models

for identifying potholes in 2D digital pictures. The analysis of road data to identify the potholes can be achieved by using a support vector machine (SVM). To find potholes, the image made use of texture measures based on histograms. Detection was made using nonlinear SVM. For the purpose of training an SVM that can recognize potholes in annotated photographs, authors in [43] employed the SIFT features they established. These methods have a detection accuracy of potholes of 91.4%. A pothole identification accuracy rate of roughly 89%was achieved by Hoang using a neural network and SVM with a steering filter-based feature extraction. By combining the SVM with the forensic-based investigation (FBI) metaheuristic, Authors in [44] achieved a 94.83% success rate in pothole detection. Experts must manually extract features to support the machine learning approach's high processing power in order to increase pothole detection accuracy. Because of the limits of their gadgets, drivers cannot take advantage of this computing capability in their own automobiles. Convolutional neural network (CNN) operations are used in deep learning approaches to automatically perform feature extraction and categorization at the same time.

III. BACKGROUND

This section presents the main methods and materials employed in the proposed methodology.

A. DIPPER THROATED OPTIMIZATION

The Dipper Throated bird, renowned for its bobbing or dipping motions while perched, belongs to the Cinclidsae family of birds. The ability of a bird to dive, swim, and hunt beneath the surface sets it apart from other passerines. It can fly straight and rapidly with no stops or glides because of its small flexible wings. The Dipper Throated bird has a distinct hunting style, quick bowing motions, and a white breast. It rushes headlong into the water to get its prey, regardless of how turbulent or fast-flowing it is. As it descends and picks up pebbles and stones, aquatic invertebrates, aquatic insects, and tiny fish perish. The great white shark uses its hands to move on the ocean floor. By bending your body at an angle and traveling down the bottom of the water with your head lowered, you might be able to locate prey. It can also dive into the water and submerge itself, using its wings to propel itself through the water and stay submerged for an extended period. The Dipper-Throated Optimization (DTO) method assumes that a flock of birds is swimming and looking for food. The following matrices can represent the position (A) and velocities (B) of the birds [45]. As previously indicated, the binary DTO is used to choose features. The continuous DTO, on the other hand, is used to improve the parameters of the classification neural network.

$$A = \begin{bmatrix} A_{1,1} & A_{1,2} & A_{1,3} & \dots & A_{1,d} \\ A_{2,1} & A_{2,2} & A_{2,3} & \dots & A_{2,d} \\ A_{3,1} & A_{3,2} & A_{3,3} & \dots & A_{3,d} \\ \dots & \dots & \dots & \dots & \dots \\ A_{m,1} & A_{m,2} & A_{m,3} & \dots & A_{m,d} \end{bmatrix} \quad (1)$$

$$B = \begin{bmatrix} B_{1,1} & B_{1,2} & B_{1,3} & \dots & B_{1,d} \\ B_{2,1} & B_{2,2} & B_{2,3} & \dots & B_{2,d} \\ B_{3,1} & B_{3,2} & B_{3,3} & \dots & B_{3,d} \\ \dots & \dots & \dots & \dots & \dots \\ B_{m,1} & B_{m,2} & B_{m,3} & \dots & B_{m,d} \end{bmatrix} \quad (2)$$

where the i^{th} bird in the j^{th} dimension is denoted by $A_{(i,j)}$ for $i \in 1, 2, 3, \dots, m$ and $j \in 1, 2, 3, \dots, d$. The bird's speed in the j^{th} dimension for $i \in 1, 2, 3, \dots, m$ and $j \in 1, 2, 3, \dots, d$ is referred to as $B_{(i,j)}$. There is a uniform distribution of the initial positions of $A_{(i,j)}$. For each bird, the values of the fitness functions $h = h_1, h_2, h_3, \dots, h_n$ are determined using the array below.

$$h = \begin{bmatrix} h_1(A_{1,1}, A_{1,2}, A_{1,3}, \dots, A_{1,d}) \\ h_2(A_{2,1}, A_{2,2}, A_{2,3}, \dots, A_{2,d}) \\ h_3(A_{3,1}, A_{3,2}, A_{3,3}, \dots, A_{3,d}) \\ \dots \\ h_m(A_{m,1}, A_{m,2}, A_{m,3}, \dots, A_{m,d}) \end{bmatrix} \quad (3)$$

where each bird's quest for food is reflected in its fitness score, the mother bird is the superior value. Sorting is done by ascending the values. A_{best} has been proclaimed the first-best solution. Normal birds A_{nd} are meant to be used as follower birds. A_{Gbest} has been named the world's best solution. The optimizer's first DTO technique for updating the swimming bird's position is based on the following equations that update the position and speed of the individuals in the population:

$$A(i+1) = \begin{cases} X & \text{if } R < 0.5 \\ Y & otherwise, \end{cases} \quad (4)$$

$$X = A_{best}(i) - K_1.|K_2.A_{best}(i) - A(i)| \quad (5)$$

$$Y = A(i) + B(i+1) \quad (6)$$

$$B(i+1) = K_3 V(i) + K_4 r_1 (A_{best}(i) - A(i)) + K_5 r_2 (A_{Gbest} - A(i)) \quad (7)$$

where i is the iteration number in which $A(i)$ is the average bird position, and $A_{best}(i)$ is the position of the best bird, and $B(i+1)$ is the bird's speed at iteration $i+1$. The K_1, K_2, and K_3 are weight values and, K_4, and K_5 are constants. The r_1 and r_2 are random values in the range [0, 1]. The steps of DTO algorithm are represented by the flowchart depicted in Fig. 3. Algorithm 1 shows the DTO algorithm step by step.

B. PARTICLE SWARM OPTIMIZATION

In particle swarm optimization (PSO), the potential solutions, referred to as particles, are flown in the search space, mimicking the intelligence of bird swarms in nature. The speed of a particle is the rate at which it changes position. The particles' positions are updated throughout time. A particle's speed is stochastically accelerated to its prior best position throughout the flight. For a particle i, the position vector is denoted by $x(i)$ and the speed vector is denoted by $y(i)$. To update the particle at iteration $t+1$ to a neighborhood of the best solution, the following equations are utilized [46].

$$x_i(t+1) = x_i(t) + y_i(t+1) \quad (8)$$

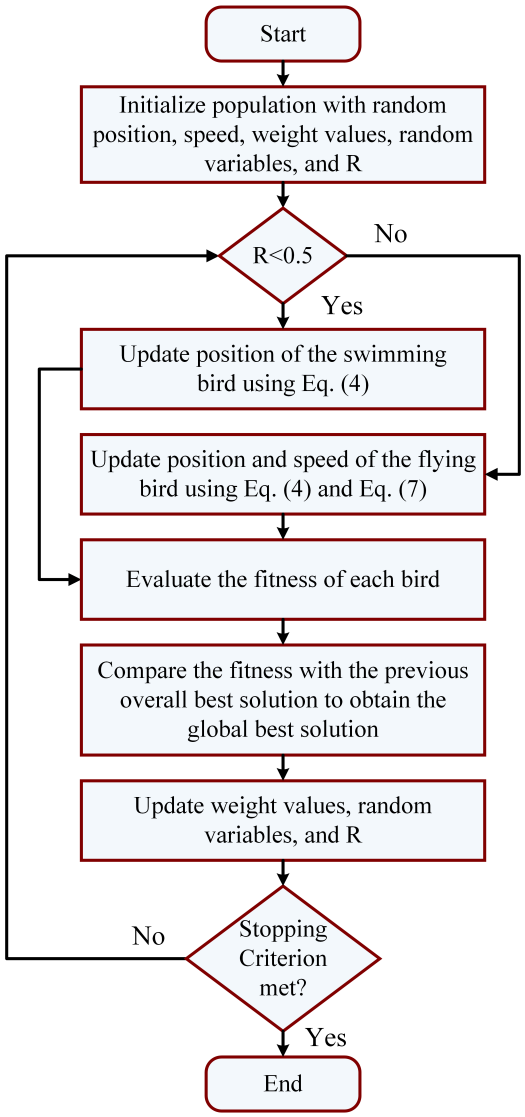


FIGURE 3. Dipper throated optimization process.

$$y_i(t+1) = y_i(t) + C_1 g_1(Pbest_i(t) + x_i(t)) + C_2 g_2(g_{best} - x_i(t)) \quad (9)$$

where g_1 and g_2 are random variables with values in the range [0, 1], and C_1 and C_2 are constants. The steps of PSO algorithm are represented by the flowchart depicted in Fig. 4.

C. GENETIC ALGORITHM

The most prevalent type of evolutionary computing is genetic algorithm (GA), which is an adaptive heuristic search and optimization algorithm inspired by the Darwinian theory [47]. GA uses the "survival of the fittest" notion to identify the optimal solution, similar to how nature progresses by mixing and choosing the best people to reproduce the next generation. The ease with which GA may be implemented while yet producing excellent results, A basic GA has five steps, according to [47].

Algorithm 1 DTO Algorithm [45]

Initialization positions of agents $A_i(i = 1, 2, \ldots, n)$ with n agents, velocities of agents $B_i(i = 1, 2, \ldots, n)$, iterations T_{max}, objective function f_n, parameters of K_1, K_2, K_3, K_4, K_5, r_1, r_2, R, $t = 1$
Calculate f_n for each agent A_i
Find best agent A_{best}
while $t \leq Tmax$ **do**
  for $(i = 1 : i < n + 1)$ **do**
    if $(R < 0.5)$ **then**
      Update position of current swimming agent as $A(i+1) = A_{best}(i) - K_1.|K_2.A_{best}(i) - A(i)|$
    else
      Update velocity of current flying agent as $B(i+1) = K_3V(i)\& + K_4r_1(A_{best}(i) - A(i)) + K_5r_2(A_{Gbest} - A(i))$
      Update position of current flying agent as $A(i+1) = A(i) + B(i+1)$
    end if
  end for
  Calculate f_n for each agent A_i
  Update K_1, K_2, R
  Find best bird A_{best}
  Set $A_{Gbest} = A_{best}$
  Set $t = t + 1$
end while
Return best agent A_{Gbest}

1) INITIALIZATION: The first phase entails the establishment of a chromosomal pool. The chromosomes are made by mixing several genes in a random order. As a result, a population with chromosomal variations emerges. The population will next go through evolution, which is the process of filtering out the weaker chromosomes while encouraging reproduction amongst the fittest chromosomes in order to get the chromosome with the "best" genes. As a result, the Population will continue to evolve from the starting Population until it reaches an end state.
2) FITNESS FUNCTION: The fitness function is a method for determining the chromosomes' strength. In order to classify the chromosomes, the evaluation of each one in the initial population, is necessary. This evaluation will foster the selection of the best individual for the next steps of the genetic algorithm process. The evaluation is based on the performance of each individual, in the following objective function:

$$F = min(error) \quad (10)$$

$$error = \sqrt{\frac{1}{n}\sum_{i=1}^{n}((X_d - X_t)/X_t)^2} \quad (11)$$

where X_d and X_t are the design variable response and the target response at instant i, respectively, and n is the total number of points.

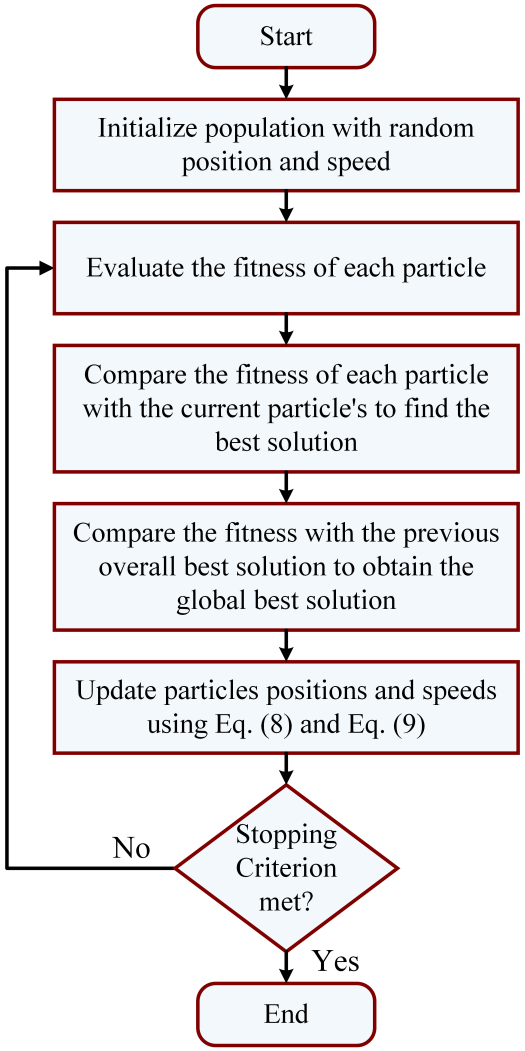


FIGURE 4. Particle swarm optimization process.

3) CROSSOVER: It is a strategy for producing kids by combining the DNA of two parents. Crossover and mutation are two strategies for generating new chromosomes from a chosen set of stronger chromosomes in the Population. As a result, the notion of parent and offspring evolved, with the children being freshly formed chromosomes.
4) MUTATION: A chromosomal change is referred to as a mutation. A chromosome can be represented in a series of binaries [0,0,0,1,1,1], for example, by viewing it in binary format. A value of 0 indicates that the gene is not chosen, whereas a value of 1 indicates that it is chosen. As a result, when the mutation occurs, the represented binary sequence flips from 0 to 1 and vice versa. The chromosome [0,0,0,1,1,1], for example, can evolve into [1,0,0,0,1,1]. To keep the GA from being stuck at the local ideal, the mutation is required. A high mutation rate, on the other hand, may make it more difficult for the GA to identify the best answer. This process is represented by fitting curve shown in Fig. 5.

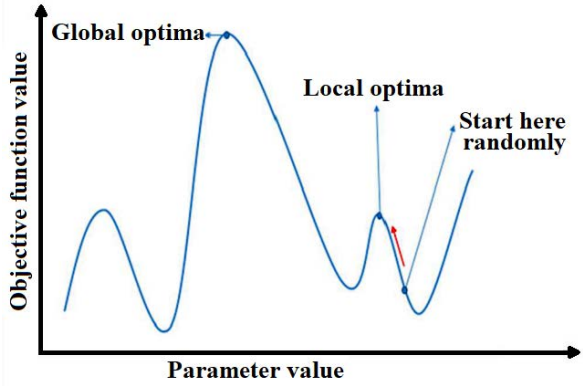


FIGURE 5. Potential values of objective function during the optimization of model parameters.

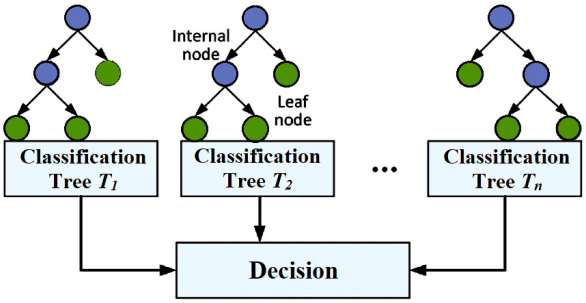


FIGURE 6. Random forest model.

5) END CONDITION: Finally, the end condition, also known as the termination criteria, is a method for determining if a GA process is converged. The number of iterations or generations can be specified here. Steps 2 through 4 make up a generation. While GA's use in search optimization is comprehensive, we're interested in a specific use case in which GA is utilized to choose the best features or to select features.

D. RANDOM FOREST MODEL

For pothole identification, random forests provide a set of advantages, although there is still room for growth. This non-parametric approach is better than the more standard statistical techniques in dealing with inadequate data [25]. Any missing data is filled up using node-specific variables. In our case, the random forest classifier enhanced the identification accuracy the most, although this isn't necessarily the case for other applications. As a result, the random forest technique linked with big data management systems can now analyze enormous datasets with numerous variables rapidly and effectively. Thus, if a class in the data is less prevalent than the others, this technique automatically balances data sets to achieve an even distribution of the data sets [48]. The variable-handling speed of this approach is well-suited to complicated tasks. There are additional advantages of using random forest, such as (i) the addition of categorized data or numeric layers is feasible as this approach is

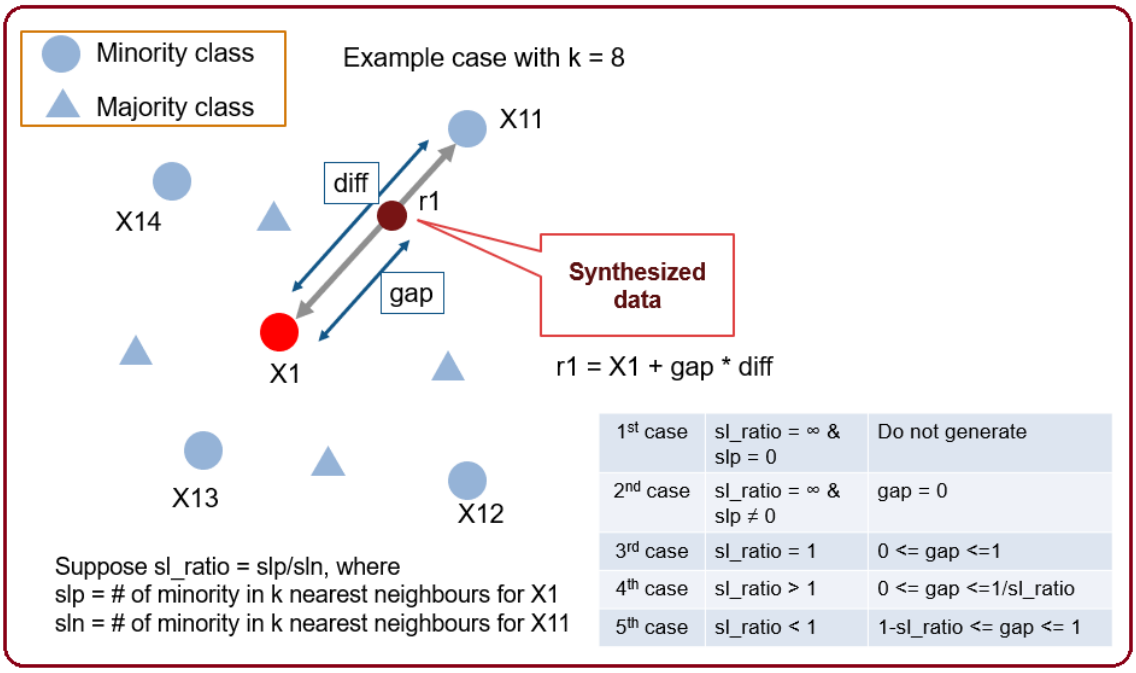


FIGURE 7. The process of synthetic minority oversampling technique (SMOTE).

non-parametric; (ii) the interpretation of rules is simple using the tree algorithm of this approach; (iii) the prerequisite to the uni-modal training data is not required; and (iv) the process of classifying potholes is cost and time-efficient when compared with the other approaches.

E. SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE

When data is imbalanced, it signifies that the prior probability of several classifications diverges greatly. In an unbalanced dataset, the majority class or negative class is named after the class with the highest prior probability, while the minority class or positive class is named after the class with the lowest prior probability. Unbalanced data can make a typical classification model perform poorly in supervised learning. The decision surface may be skewed in favor of the majority class, resulting in low minority class classification accuracy. Many real-world problems have unbalanced data as the sources of imbalance might be numerous for a practical situation [49].

K-fold cross-validation is one of the most effective methods for determining the most accurate machine-learning methodology [50]. For both classification and regression problems, if the dataset is significantly unbalanced, the model will strive to modify itself to better model the more populated class. Consequently, the model performance of the low-population classes will be poor. Two approaches have been developed to overcome this limitation: (1) adding new instances of low-populated classes or deleting instances of the most populated classes, and (2) using metrics to evaluate the model's performance that weight the performance of different classes or value ranges to avoid the effect of the dataset imbalance.

For the datasets of small size, only one method is available to artificially balance it: by producing new instances of the low-represented classes. This is the most common strategy in problems with small or medium datasets, such as the one presented in this research. On the other hand, reduction of instances is primarily used in big data problems, such as failure detection, due to a large number of instances of the majority class, such as the non-failure class. SMOTE is a well-known approach for generating additional synthetic samples in a dataset's minority class [51]. It takes one member of the minority class and chooses one of its k closest neighbors at random, where k is a SMOTE algorithm parameter. The new instance is constructed at a random location along the segment connecting the selected neighbor and the original minority-class instance. SMOTE's procedure is shown in Figure 7. It's worth noting that in this implementation, SMOTE only duplicates one class (the least-populated one) in order to increase the number of instances in that class. As a result, if many classes have a low number of instances, the SMOTE technique should be employed in more than one iteration, and if the duplication of the number of instances is insufficient to balance the dataset, more than one iteration should be performed on the same class.

IV. THE PROPOSED METHODOLOGY

The proposed methodology for detecting potholes is depicted in Fig. 8. As shown in the figure, the methodology starts with extracting a set of relevant features from the input image using the ResNet-50 deep network. The most significant features are selected using a new proposed binary particle swarm/dipper throated optimizer (bPSDTO). The selected features are then balanced using the proposed optimized

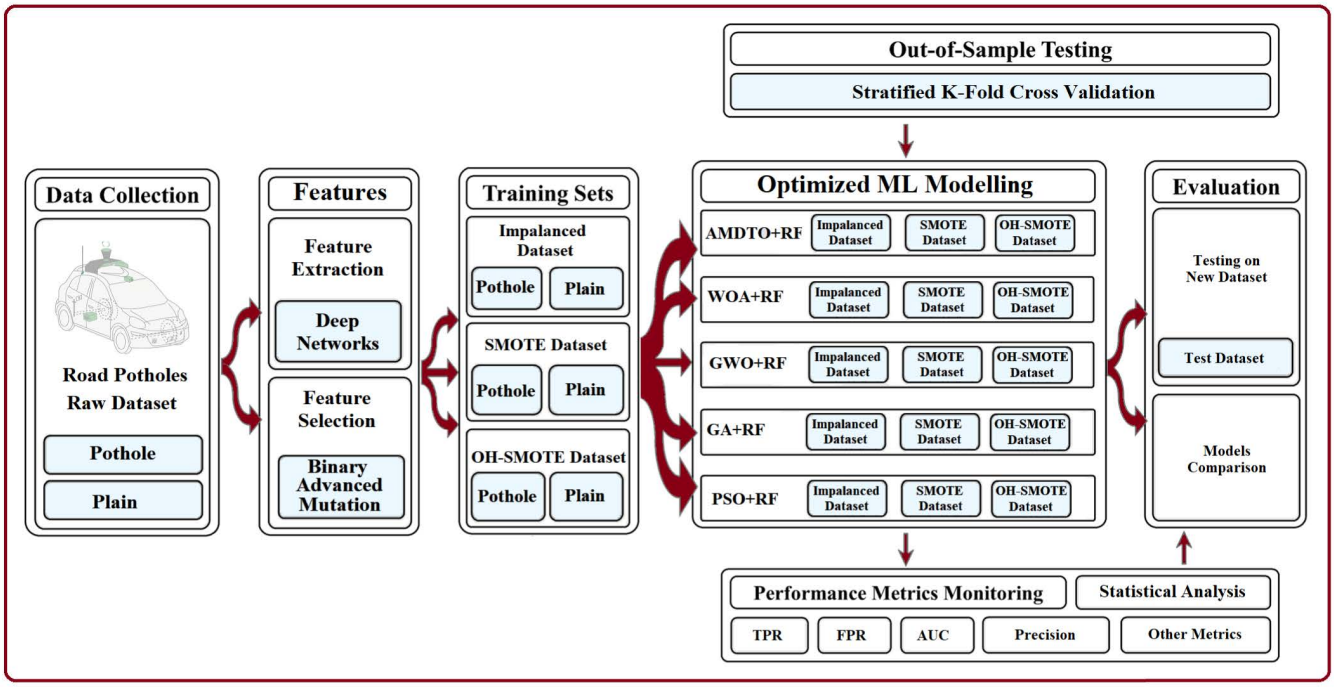


FIGURE 8. The architecture of the proposed methodology.

TABLE 3. Data augmentation of the pothole dataset.

No.	Augmentation method
1	Inverse all pixels
2	Apply Gaussian kernel to blur image
3	Random values in the range -40 to 40 are added to all pixels
4	Medians of neighborhood pixels are used to blur image
5	Average of neighborhood pixels is used to blur image
6	Translate image from -30 to 30 pixels on x-axis and y-axis, and change image size to 70-130 pixels
7	Kernel of size 3x3 is Convolved with image
8	Convert 2% of all pixels to black pixels
9	Merge the original image with an edge binary of the image
10	Convert p% of all pixels to black pixels ($0 \leq p \leq 2$)
11	merge the original image with the embossed image
12	Convert an image into a Superpixels partial representation
13	Addition of pepper noises at 5% of all pixels
14	Multiplying pixels with a random value between 0.5 and 1.5 to Covert them
15	Flip images horizontally

hashing SMOTE-based algorithm. The balanced dataset is then used to train a random forest model, which is optimized using the proposed advanced mutation dipper throated optimization (AMDTO) to classify the input image. More details about the steps of the proposed methodology are presented in the next sections.

A. DATA AUGMENTATION

Pothole and plain road images are augmented using various methods of data augmentation. The augmentation operations are listed in Table 3. The table shows that fifteen operations were applied to the input images to augment the dataset. These operations include changing pixel colors, applying various transformations and convolution operations, and adding noise fractions. Due to the limited size of the dataset, we applied the fifteen data augmentation operations to increase the number of images in the dataset. The dataset after augmentation contains 15, 000 images, 12, 000 pothole road images and 3, 000 plain road images. The augmented dataset is then fed into the feature extraction process, and the proposed balancing operation is applied, as discussed in the next sections.

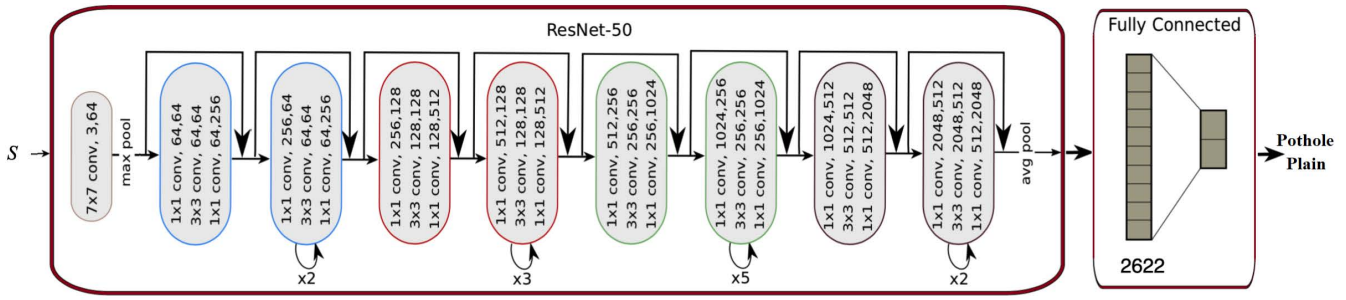


FIGURE 9. The architecture of ResNet-50 deep network used in feature extraction.

B. FEATURE EXTRACTION

The extraction of the features that accurately represent the pothole and plain road is performed in terms of the ResNet-50 deep neural network, which is pre-trained on the ImageNet dataset. The architecture of the ResNet-50 model is shown in Fig. 9. The ResNet50 model consists of 49 convolutional layers plus a fully-connected layer with 16 residual blocks [52]. In the residual bottleneck block, the network model of "Shortcut Connections" may integrate the learned shallow structure with its mapping's adding layer. The two sections are joined, the same as fusing the feature information from the bottom layer into the higher layer. As a result, such a network topology can preserve the vital information of select tiny receptive field targets while preventing deep neural network performance deterioration and gradient disappearance. The bottleneck structure's network building blocks, on the other hand, might increase time complexity and model size while lowering model training efficiency. ResNet50, in comparison to AlexNet and VGG16, stacks residual blocks and addresses network degradation to extract more in-depth voiceprint characteristics and improve the network's fitting capabilities. Figure 6 depicts the network model. The relevant features are extracted using the convolutional layer conv1 and four types of residual blocks conv2 conv5. The output of the convolution is then flattened by the layers that are completely linked. The resulting features are then fed to the proposed optimization algorithm to select the most significant set of features through the feature selection process, which is discussed in the next section.

C. THE PROPOSED OPTIMIZATION ALGORITHM

Finding global optima is a difficult target to find. For the suggested method, two efficient methods are described. The PSO is the first algorithm in which individuals are moved based on their local and global optimal placements. The position of the best global individual refers to the best position found by the whole population, whereas the local best position refers to an individual's best position thus far. Individuals in PSO can converge on their global goals thanks to this social behavior. Nature's flock of birds and fish school has an impact on their behavior. We chose PSO for our proposed hybrid optimizer due to its simplicity, dependability, and strength. In the proposed hybrid approach, the second optimizer is DTO, a swarm-based meta-heuristic optimizer that mimics the social hierarchy and foraging behavior of dipper throated birds. The position and speed of the bird agents affect how individuals travel in the DTO. In the proposed hybrid optimizer, the optimization process starts with a group of random individuals. Candidate solutions to the problem being solved have been suppressed by such individuals. For the initial solution and at each iteration, the fitness function is computed for all individuals. As previously mentioned, two groups represent the population, the first of which follows the PSO method and the second of which follows the DTO process. Consequently, the search space is thoroughly examined for potential spots, then exploited utilizing the strong DTO and PSO algorithms. The steps of the proposed PSDTO algorithm pseudo-code are shown in Fig. 10.

D. THE PROPOSED BINARY PSDTO (bPSDTO)

The selection of relevant features is naturally a binary problem. Therefore, a modification to the continuous solutions resulted from the proposed PSDTO to adapt to the problem of feature selection. The basic idea of this modification is to convert the continuous solutions retrieved by PSDTO to binary solutions. In specific, the position and speed of agents in the search space are changed continuously in the proposed PSDTO. However, these movements are changed in binary for feature selection.

According to the update equations of the speed and position of the agents in the proposed algorithm in equations (5, 6, 7, 8, and 9), the updated position the birds in the search space is performed using the following equation.

$$A_b(i+1) = \begin{cases} 1 & \text{if } Sigmoid(m) \geq rand \\ 0 & otherwise, \end{cases} \tag{12}$$

where $A_b(i+1)$ is the updated binary position at iteration $t+1$, and *rand* is a number selected randomly from a uniform distribution. The speed of the birds as well as the position and speed of the particles are updated similarly. $Sigmoid(m)$ function is defined as follows.

$$Sigmoid(m) = \frac{m}{1 + e^{-10(m-0.5)}} \tag{13}$$

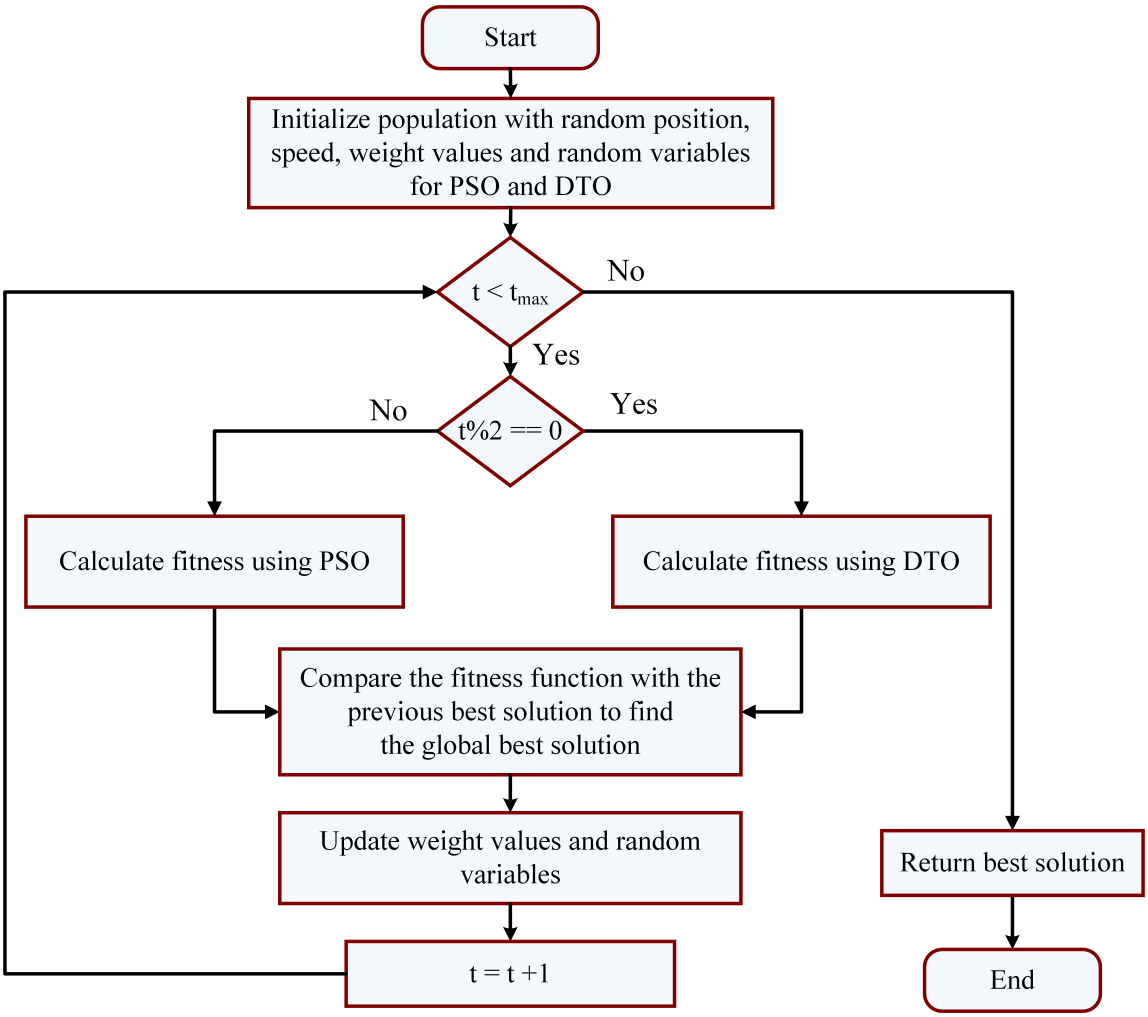


FIGURE 10. The proposed hybrid PSDTO algorithm.

The steps of the proposed binary optimizer for feature selection is presented in Algorithm 2.

E. FITNESS FUNCTION

A fitness function is used to measure the quality of each hybrid PSDTO solution. The fitness function is influenced by the classification error rate and the number of provided features. The solution is considered great if it identified a subset of characteristics that resulted in a lower classification error rate and fewer selected features. To determine the quality of each solution, apply the following formula:

$$Fitness = v_1 Error + v_2 \frac{|S|}{|T|} \tag{14}$$

where *Error* represents the error rate of the KNN classifier. The length of the selected features is denoted by $|S|$. The length of the full features is referred to as $|T|$. The factors v_1 and v_2 are in the range of $[0, 1]$, where $v_1 = 1 - v_2$.

F. THE PROPOSED DATASET BALANCING METHOD

The recently emerged SMOTE approach based on locality sensitive hashing (LSH), referred to as LSH-SMOTE, has evolved for effectively and swiftly locating the nearest neighbors of the minority class. The dataset is hashed and divided into buckets in this approach, with comparable objects with identical hash codes being allocated to the same bucket, increasing the likelihood of collision and making the search for the k-nearest neighbors in each bucket easier. The most colliding instances from each bucket are then picked. The Euclidean distance is then used to sort the instances, and only the K-nearest neighbors colliding instances to the query instance are selected. Finally, the main SMOTE class receives a list containing the instances of the k-nearest neighbors, which are used to generate synthetic instances. Because the LSH complexity is O (d log n) versus O (d n) for linear searches like (K-NN), combining the LSH approach with the SMOTE algorithm will result in a significant time reduction. Unbalanced data can make a typical classification model perform poorly in supervised learning. To achieve more balance

Algorithm 2 The Proposed Binary PSDTO Algorithm

Initialize PSDTO parameters.
Convert best solutions to binary [0,1].
Evaluate fitness of the current solutions
Train KNN and estimate the fitness
while $t \leq Iter_{max}$ **do**
  Apply PSDTO algorithm to get best solutions
  Convert best solutions to binary using:

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

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

  Calculate Fitness
  Update PSDTO parameters
  Update Positions and speeds of the best solutions
  Update t = t + 1
end while
Return best solution

between the majority and minority classes, we propose two enhancements to the recently published LSH-SMOTE algorithm. In this section, the proposed enhancements are presented and discussed.

The proposed method is shown in Fig. 11. As shown in the figure, the minority dataset is split into a number of N groups; each group contains a list of instances from the dataset. The assignment of instances from the minority class to these groups is achieved through the application of the proposed PSDTO optimization algorithm. In specific, the

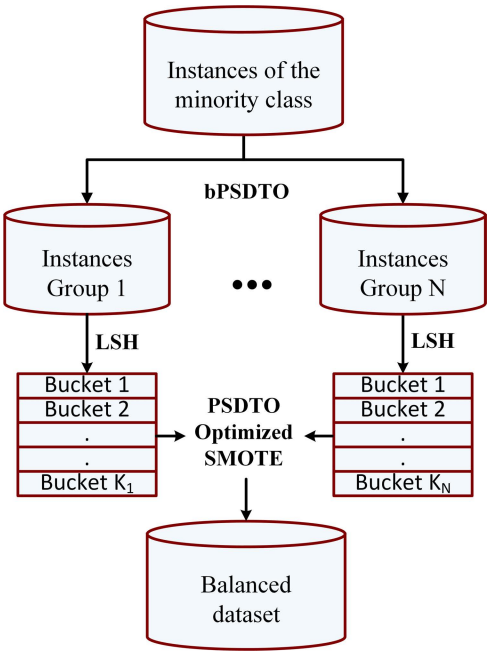


FIGURE 11. The proposed dataset balancing approach.

Algorithm 3 The Proposed Optimized SMOTE Algorithm

Input P instances of a bucket form LSH buckets
Optimize KNN parameters using PSDTO
while $x \in P$ **do**
  Find K-nearest neighbors to x in P
  Find y by randomizing one of the K instances
  Calculate $difference = x - y$
  Pick gap as a random number between 0 and 1
  Calculate $n = x + difference * gap$
end while
Return new instances

instances are dealt with as feature instances; the top most significant instances are located in the first group, then the optimizer is applied to the remaining samples to select the top most significant instances to be located in the second group, and so on. Once the groups are filled with instances from the minority class, the LSH method is applied to allocate the instances of each group into a set of buckets based on a hash function and the resulting collisions. The advantage of the proposed approach is that it can be easily run in parallel, which can greatly reduce the time needed to balance datasets of huge sizes.

The steps of the proposed optimized SMOTE are presented in Algorithm 2. In this algorithm, the process starts with receiving a bucket from the buckets generated by the LSH method. On the other hand, based on the instances of the received bucket, the K value of the KNN algorithm is optimized using the proposed PSDTO. The instances of the received bucket are then processed by moving over each instance and finding its neighbors; then, the difference between the instance under consideration and a random instance from its K neighbors is calculated. The resulting difference is used to find the value of the newly generated instance, which is then added to the balanced dataset. Balanced data can make a typical classification model perform better.

G. THE PROPOSED ADAPTIVE MUTATION DTO

Once the dataset is balanced, and the features are selected properly, the random forest classifier is employed to classify the input features. However, in this work, we developed a new optimization algorithm for optimizing the parameters of the random forest to boost classification accuracy. The proposed optimization algorithm is based on adaptive mutation of the genetic algorithm, which is applied in conjunction with the DTO algorithm. Fig. 12 depicts the process of the proposed approach. As shown in the figure, the process starts with randomly generating the initial population of dipper-throated birds, then calculating the fitness of each bird to select the best private and global solutions. The population is then split into two groups; the first group is searched using a genetic algorithm, whereas the second group is searched using the DTO algorithm. Based on the results of the two

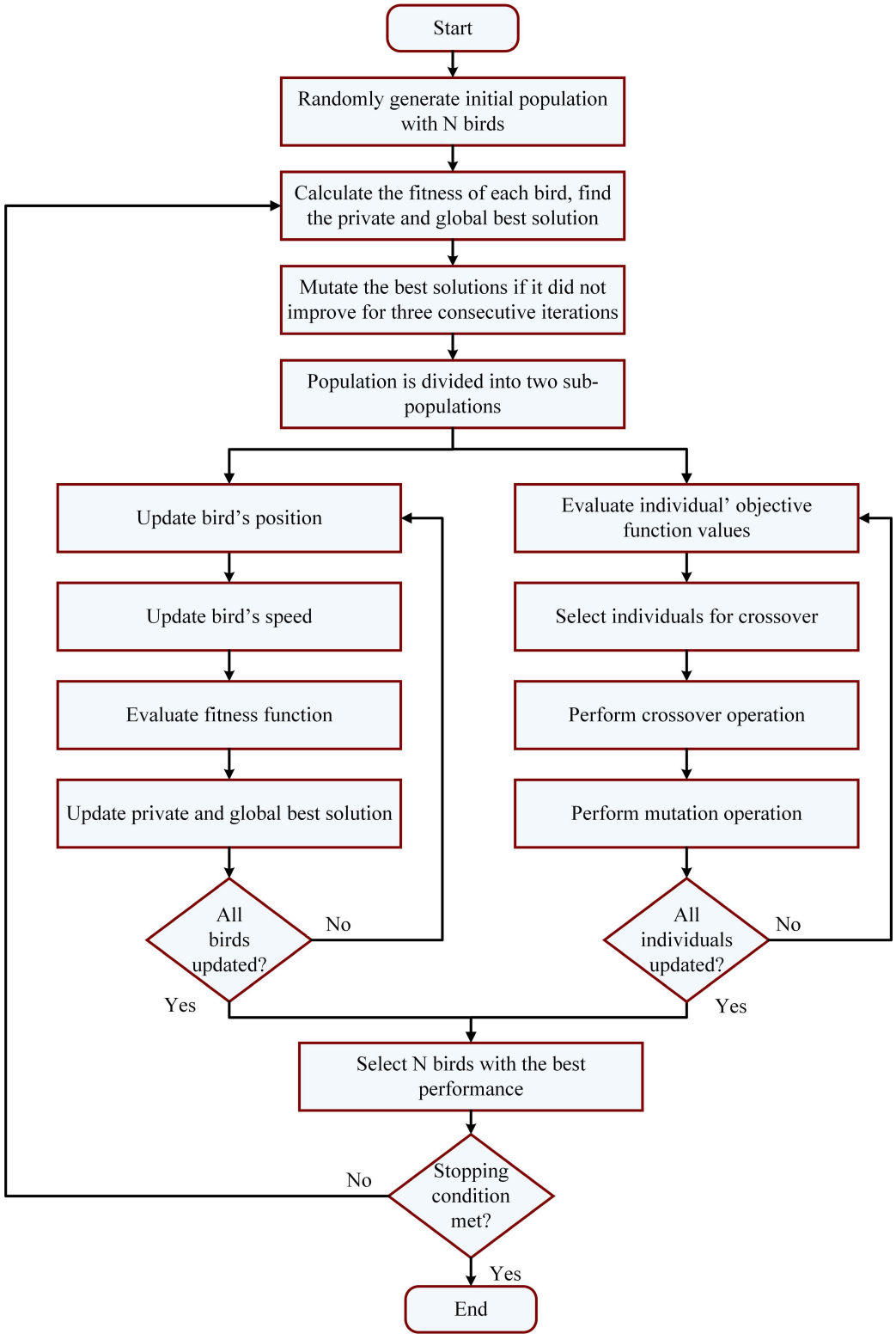


FIGURE 12. The process of the proposed adaptive-mutation-DTO (AMDTO).

algorithms, the best N solutions were selected. This process is repeated in iterations until the stopping condition is met. During the iterations, if the best solution did not improve for three consecutive iterations, the best solutions are mutated to increase the exploration operation of search space and thus given a higher potential to reach the optimal solution. This

Algorithm 4 Proposed AMDTO Algorithm

Initialization positions of agents $A_i(i = 1, 2, \ldots, n)$ with n agents, velocities of agents $B_i(i = 1, 2, \ldots, n)$, iterations T_{max}, objective function f_n, parameters of $k, K_1, K_2, K_3, K_4, K_5, r_1, r_2, R, t = 1$
Calculate f_n for each agent A_i
Find best agent A_{best}
while $t \leq Tmax$ **do**
  Split Solutions to GA group (n_1) and DTO group (n_2)
  for $(i = 1 : i < n_1 + 1)$ **do**
    Apply GA Crossover Process
    Apply GA Mutation Process
  end for
  for $(i = 1 : i < n_2 + 1)$ **do**
    if $(R < 0.5)$ **then**
      Update position of current swimming agent as $A(i+1) = A_{best}(i) - K_1.|K_2.A_{best}(i) - A(i)|$
    else
      Update velocity of current flying agent as $B(i+1) = K_3V(i)\& + K_4r_1(A_{best}(i) - A(i)) + K_5r_2(A_{Gbest} - A(i))$
      Update position of current flying agent as $A(i+1) = A(i) + B(i+1)$
    end if
  end for
  Calculate f_n for each agent A_i
  Find best N agents
  Set $A_{Gbest} = A_{best}$
  if (Best f_n is same for three iterations) **then**
    Apply the following mutation equation
    $$A(i) = 1 - \frac{2 * k * A(i) * i^2}{|A(i)|}$$
  end if
  Update $k, K_1, K_2, R, t = t + 1$
end while
Return best agent A_{Gbest}

mutation is performed using the following equation.

$$A(i) = 1 - \frac{2 * k * A(i) * i^2}{|A(i)|} \tag{15}$$

where $A(i)$ is the best bird position at iteration i, k is a number with values decreasing exponentially from 1 and 0, and $|A(i)|$ is the number of best solutions found at iteration i. Algorithm 4 shows the proposed AMDTO algorithm in detail.

H. THE PROPOSED BINARY AMDTO (bAMDTO)

To employ the proposed AMDTO algorithm for feature selection, it should be modified to adapt to this target. In this section, we proposed a binary version of the proposed AMDTO in which the continuous solution is converted to binary (0 or 1), where 0 means the feature is not selected and 1 means the feature is selected. The steps of the proposed binary AMDTO (bAMDTO) are presented in Algorithm 5.

Algorithm 5 The Proposed Binary AMDTO Algorithm

Initialize AMDTO parameters
Convert best solutions to binary [0,1]
Evaluate fitness of the current solutions
Train KNN and estimate the fitness
while $t \leq Iter_{max}$ **do**
  Apply AMDTO algorithm to get best solutions
  Convert best solutions to binary using:
$$S_b^{(t+1)} = \begin{cases} 1 & \text{if } Sigmoid(m) \geq 0.5 \\ 0 & otherwise \end{cases},$$
$$Sigmoid(m) = \frac{1}{1 + e^{-10(m-0.5)}}$$
  Calculate Fitness
  Update AMDTO parameters
  Update Positions and speeds of the best solutions
  Update t = t + 1
end while
Return best solution

Algorithm 6 The Proposed Optimized RF Classifier

Initialize AMDTO parameters.
Initialize number of trees B.
while $b \in B$ **do**
  Draw a bootstrap sample S^* of size N from the training data
  Grow a random forest tree T_b to the boostrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached:
  (i) Select m variables using the proposed AMDTO algorithm
  (ii) Pick the best variable/split-point among the m
  (iii) Split the node into two children nodes
end while
Output the ensemble of trees $\{T_b\}^B$
Let $C_b(x)$ be the class prediction of the b^{th} random forest tree
Return majority vote $\{C_b(x)\}^B$

I. THE PROPOSED OPTIMIZED RF CLASSIFIER

To achieve better classification performance, the proposed AMDTO algorithm is employed to optimize the parameters of random forest (RF) classifier. Algorithm 6 shows the steps of the proposed optimized RF algorithm. As shown in the algorithm, the optimization is performed to select the best split among the optimized variable m.

J. STRATIFIED K-FOLD CROSS VALIDATION

Cross-validation (CV) is a common practice followed to improve the generalization of the trained machine learning models. The operation of CV is based on separating the dataset into equal parts, known as folds. Therefore, if the

FIGURE 13. Samples images of pothole and plain roads.

number of folds is five, then the dataset is divided into five equal halves, training is done using four-folds, and testing is done with one fold. After that, one fold from training is sent to testing, and the test set is passed to the training set. This process is continued until all potential combinations have been exhausted, guaranteeing that all data has been trained and tested at least once, allowing the model's generalization to be evaluated. On the hand, each fold is stratified to ensure containing samples representing all the classes of the given task.

V. EXPERIMENTAL RESULTS

In this work, three experiments were conducted to assess the proposed algorithms. The first experiment is conducted to investigate the efficiency of the deep learning networks used in feature extraction to choose the best network. The second experiment assesses the effect of the proposed dataset balancing approach based on optimized hash SMOTE and PSDTO algorithm. The third experiment is conducted to evaluate the performance of the proposed AMDTO algorithm.

A. DATASET

The pothole dataset employed in this work is available publicly on Kaggle [53]. The dataset consists of 700 images of two classes, namely, pothole and plain. To train the feature extractor and evaluate the proposed method, the dataset is split into 80% for (training and validation), and the remaining 20% is dedicated to testing. Sample images from the dataset are presented in Fig. 13. Due to the limited size of this dataset, we applied the aforementioned data augmentation operations to increase the number of images in the dataset. The dataset after augmentation contains 15, 000 images, 12, 000 pothole road images and 3, 000 plain road images.

B. EVALUATION CRITERIA

The achieved results of the proposed approaches are evaluated in terms of the performance metrics presented in Table 4. In this table, the first set of metrics is used to measure the performance of the feature selection process. In contrast, the second set of metrics is used to measure the performance of the classification using the proposed optimized random forest. In the table, the number of runs of an optimizer is indicated as M, the best solution at the run number j is denoted by g_j^*, $size(g_j^*)$ refers to the size of the best solution vector. N denoted the number of points in the test set, and C_i refers to the output label results from the employed classifier for the data point i. L_i is the label of the class of the point i, and D refers to the total number of features. TP, TN, FP, and FN refer to the true positive, true negative, false positive, and false negative.

TABLE 4. Evaluation metrics of the models' performance.

Metric	Value
Mean	$\frac{1}{M}\sum_{i=1}^{M} g_*^i$
Best Fitness	$min_{i=1}^{M} g_*^i$
Worest Fitness	$max_{i=1}^{M} g_*^i$
Average fitness size	$\frac{1}{M}\sum_{i=1}^{M} size(g_*^i)$
Average Error	$\frac{1}{M}\sum_{j=1}^{M}\frac{1}{N}\sum_{i=1}^{N} mse(C_i, L_i)$
Standard deviation	$\sqrt{\frac{1}{M-1}\sum_{i=1}^{M}(g_*^i - Mean)^2}$
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$
Pvalue (PPV)	$\frac{TP}{TP+FP}$
Nvalue (NPV)	$\frac{TN}{TN+FN}$
Specificity (TNR)	$\frac{TN}{TN+FP}$
Sensitivity (TPR)	$\frac{TP}{TP+FN}$
F1-Score	$\frac{TP}{TP+0.5(FP+FN)}$

C. FEATURE EXTRACTION RESULTS

The extraction of significant features from the input images is a critical process that affects the performance of machine learning models. In this work, we tested four deep neural networks namely, AlexNet [54], VGG-19 [55], GoogleNet [56],

TABLE 5. Performance of the deep neural networks used in feature extraction.

	AlexNet	VGG-19	GoogLeNet	ResNet-50
Accuracy	86.8%	87.3%	90.4%	93.8%
TPR	0.625	0.625	0.740	0.789
TNR	0.955	0.951	0.955	0.968
PPV	0.833	0.800	0.833	0.833
NPV	0.876	0.889	0.922	0.957
F1-score	0.714	0.702	0.781	0.811
Time (S)	677	599	475	312

and ResNet-50 [57] for extracting features from the given dataset. The performance of these networks is presented in Table 5. In this experiment, the default parameters are chosen because the first stage is used to retrieve images' features from earlier layers of a deep neural network for use in subsequent operations for feature selection and balancing. As shown in the table, the network that achieved the best results was ResNet-50. Therefore, we adopted this network for feature extraction in this work.

D. DATASET BALANCING RESULTS

In this experiment, we will compare the results achieved by the random forest classifier using the selected features based on three cases 1) without dataset balancing (Imbalanced), 2) with dataset balancing using the original LSH-SMOTE algorithm [51], and 3) using the proposed optimized SMOTE approach. The achieved results are evaluated in Table 6. It can be noted from this table that the results achieved by the proposed dataset balancing approach outperform those values achieved by the imbalanced and the LSH-SMOTE-based approach. The achieved accuracy using the proposed approach is (96.4%) within less time, whereas the achieved accuracy by the imbalanced and LSH-SMOTE approaches are (94.4%) and (95.5%), respectively. The balanced dataset using the proposed method is used to train the optimized random forest classifier, and the results are discussed in the next section.

TABLE 6. Evaluation results of the results achieved using the data balancing approaches.

	SMOTE	LSH-SMOTE	PSDTO SMOTE
Accuracy	0.944	0.955	0.964
TPR	0.818	0.818	0.818
TNR	0.977	0.982	0.986
PPV	0.911	0.921	0.913
NPV	0.955	0.965	0.973
F1-score	0.857	0.857	0.857
Time (S)	211	178	92

E. FEATURE SELECTION RESULTS

The selection of the most significant features from the features extracted by ResNet-50 is essential. In this experiment, twelve optimization methods were employed; namely, proposed bAMDTO, binary gray wolf optimizer (bGWO) [58], binary particle swarm optimizer (bPSO) [46], bGWOPSO [59], binary genetic algorithm (bGA) [47], bGWOGA [60], binary bat algorithm (bBA) [61], binary whale optimization algorithm (bWOA) [62], binary biogeography optimization (bBBO) [63], binary Multiverse Optimization (bMVO) [64], binary Satin Bowerbird Optimizer (bSBO) [65], and binary Firefly Algorithm (bFA) [66]. Table 7 shows the assessment of the results achieved by these optimization methods. As presented in this table, the proposed bAMDTO algorithm achieves the best results when compared to other methods.

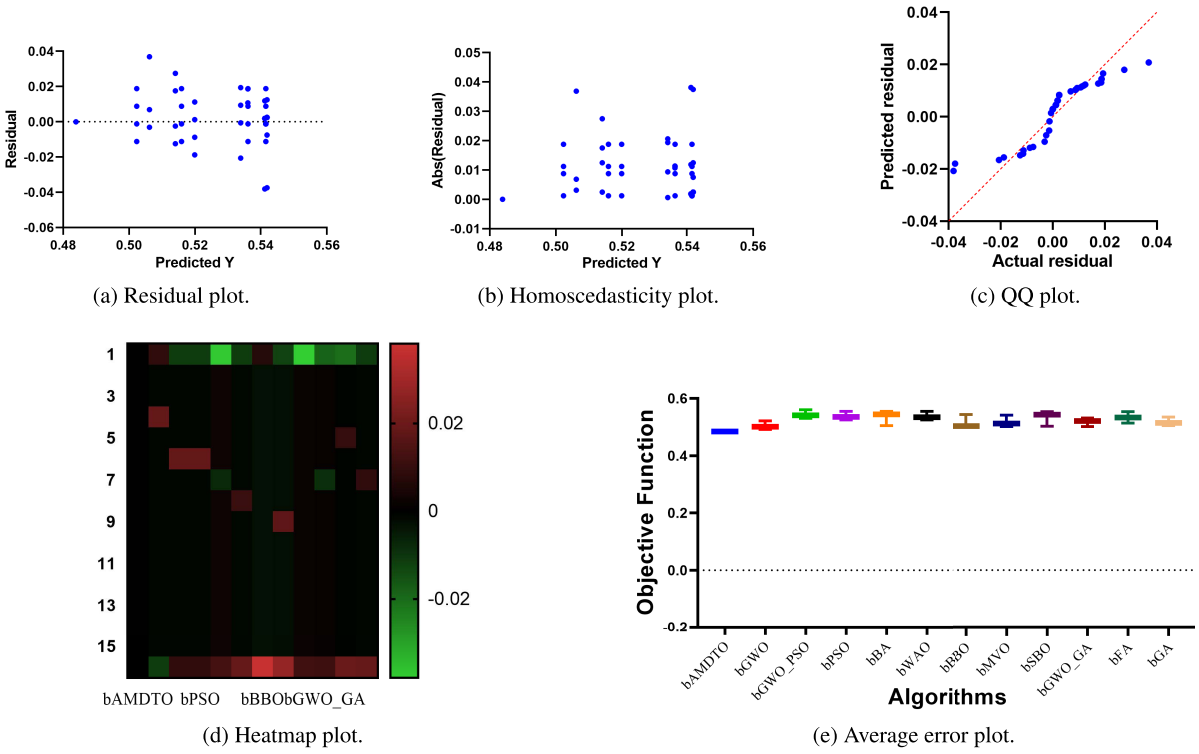
The p-values between the proposed feature selection algorithm (bAMDTO) and the other competing algorithms are calculated using the statistical difference between every two algorithms to show that the suggested approach is significantly different. Wilcoxon's rank-sum test is employed to do this analysis. The null and alternative hypotheses are two primary hypotheses in this test. The mean, values of null hypothesis represented by $H0$ includes bAMDTO = bGWO, bAMDTO = bGWO-PSO, bAMDTO = bPSO, bAMDTO = BA, bAMDTO = bWOA, bAMDTO = bBBO, bAMDTO = bMVO, bAMDTO = bSBO, bAMDTO = bGWO-GA, bAMDTO = bFA, and bAMDTO = bGA. On the other hand, $H1$ hypothesis does not consider the algorithms' means in the comparison. Table 8 presents the results of the Wilcoxon rank-sum test. The p-values between the proposed algorithm and the other algorithms are less than 0.05. These results prove the superiority and statistical significance of the proposed feature selection method.

The statistical difference between the proposed bAMDTO algorithm and the other algorithms is investigated using the one-way analysis of variance (ANOVA) test. The mean, values of the null hypothesis, designated by $H0$, include bAMDTO = bGWO = bGWO-PSO = bPSO = BA = bWOA = bBBO = bMVO = bSBO = bGWO-GA = bFA = bGA. Table 9 presents the measured values of the ANOVA test.

A visual representation of the analysis performed on the achieved results using based on the proposed feature selection algorithm is shown in Fig. 14. The first three plots represent the residual, homoscedasticity, and QQ plots. As shown in these plots, the residual error falls in the range of -0.02 and $+0.02$, and the values of homoscedasticity lay in the range -0.01 and $+0.04$, which reflect the robustness of the proposed approach. In addition, the QQ plot shows that the predicted results match the actual values, which confirms the robustness of the proposed approach. On the other hand, the last two plots, the heat map and average error plots show the comparison between the results achieved by the proposed feature selection method and the other competing feature selection methods in the literature. The results shown in these plots clarify the superiority of the proposed method.

F. OPTIMIZED RANDOM FOREST RESULTS

The classification of potholes is performed in terms of the features selected using the proposed bAMDTO. The adopted



(a) Residual plot. (b) Homoscedasticity plot. (c) QQ plot.

(d) Heatmap plot. (e) Average error plot.

FIGURE 14. Analysis of the results achieved by the proposed feature selection algorithm.

TABLE 7. Performance measurement of the feature selection algorithms.

	Avg. error	Avg. Select size	Avg. Fitness	Best Fitness	Worst Fitness	STD Fitness
bAMDTO	0.48388	0.43668	0.54708	0.44888	0.54738	0.36938
bGWO	0.50108	0.63668	0.56328	0.48358	0.55048	0.37408
bPSO	0.53488	0.63668	0.56168	0.54198	0.60968	0.37348
bGWOPSO	0.54038	0.76998	0.57158	0.52508	0.63508	0.39228
bGA	0.51468	0.57908	0.57468	0.47798	0.59308	0.37568
bGWOGA	0.52118	0.55948	0.56938	0.54718	0.62338	0.37468
bBA	0.54448	0.77608	0.58458	0.47428	0.57588	0.38338
bWAO	0.53468	0.80008	0.56948	0.53358	0.60968	0.37568
bBBO	0.50308	0.80048	0.56738	0.55708	0.64358	0.41838
bMVO	0.51158	0.73318	0.59138	0.51658	0.63458	0.42418
bSBO	0.54318	0.80698	0.60138	0.54448	0.62418	0.43438
bFA	0.53328	0.67118	0.61358	0.53228	0.62988	0.41028

TABLE 8. Wilcoxon signed-rank test of the results achieved by the feature selection algorithms.

	bAMDTO	bGWO	bGWO_PSO	bPSO	bBA	bWAO	bBBO	bMVO	bSBO	bGWO_GA	bFA	bGA
Theoretical median	0	0	0	0	0	0	0	0	0	0	0	0
Actual median	0.4839	0.5011	0.5404	0.5349	0.5445	0.5347	0.5031	0.5116	0.5432	0.5212	0.5333	0.5147
Number of values	16	16	16	16	16	16	16	16	16	16	16	16
Wilcoxon Signed Rank Test												
Sum of signed ranks (W)	136	136	136	136	136	136	136	136	136	136	136	136
Sum of positive ranks	136	136	136	136	136	136	136	136	136	136	136	136
Sum of negative ranks	0	0	0	0	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	<0.0001	<0.0001	<0.0001	<0.0001
Exact or estimate?	Exact	Exact	Exact	Exact	Exact	Exact	Exact	Exact	Exact	Exact	Exact	Exact
P value summary	****	****	****	****	****	****	****	****	****	****	****	****
Significant (alpha=0.05)?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
How big is the discrepancy?												
Discrepancy	0.4839	0.5011	0.5404	0.5349	0.5445	0.5347	0.5031	0.5116	0.5432	0.5212	0.5333	0.5147

classifier is the random forest (RF). To achieve the best performance using RF classifier, we optimized its parameters using the proposed AMDTO algorithm. A set of experiments was conducted to evaluate the performance of the optimized RF classifier. To show the superiority of the proposed optimization of RF, a comparison between the performance

TABLE 9. One-way analysis of variance (ANOVA) test of the feature selection algorithms.

	SS	DF	MS	F (DFn, DFd)	P-value
Treatment (between columns)	0.06212	11	0.005647	F (11, 180) = 96.14	P<0.0001
Residual (within columns)	0.01057	180	0.00005874		
Total	0.07269	191			

TABLE 10. Analysis of the results achieved by the proposed AMDTO+RF approach and the other approaches.

	AMDTO+RF	WOA+RF	GWO+RF	GA+RF	PSO+RF
Number of values	19	19	19	19	19
Minimum	0.9955	0.9695	0.976	0.9695	0.9781
Maximum	0.9985	0.985	0.9916	0.985	0.991
Median	0.9965	0.975	0.986	0.975	0.981
Range	0.003	0.0155	0.0156	0.0155	0.0129
25% Percentile	0.9965	0.975	0.986	0.975	0.981
75% Percentile	0.9985	0.985	0.9916	0.985	0.991
10% Percentile	0.9965	0.975	0.986	0.975	0.981
90% Percentile	0.9985	0.985	0.9916	0.985	0.991
95% CI of median					
Upper confidence limit	0.9985	0.985	0.9916	0.985	0.991
Lower confidence limit	0.9965	0.975	0.986	0.975	0.981
Actual confidence level	98.08%	98.08%	98.08%	98.08%	98.08%
Std. Deviation	0.001017	0.005131	0.003783	0.005131	0.004926
Std. Error of Mean	0.0002334	0.001177	0.000868	0.001177	0.00113
Mean	0.9971	0.9779	0.9873	0.9779	0.984
Lower 95% CI of mean	0.9966	0.9754	0.9855	0.9754	0.9816
Upper 95% CI of mean	0.9976	0.9803	0.9892	0.9803	0.9864
Coefficient of variation	0.1020%	0.5247%	0.3832%	0.5247%	0.5006%
Geometric mean	0.9971	0.9779	0.9873	0.9779	0.984
Geometric SD factor	1.001	1.005	1.004	1.005	1.005
Upper 95% CI of geo. mean	0.9976	0.9803	0.9892	0.9803	0.9864
Lower 95% CI of geo. mean	0.9966	0.9754	0.9855	0.9754	0.9816
Harmonic mean	0.9971	0.9778	0.9873	0.9778	0.984
Upper 95% CI of harm. mean	0.9976	0.9803	0.9892	0.9803	0.9864
Lower 95% CI of harm. mean	0.9966	0.9754	0.9855	0.9754	0.9816
Quadratic mean	0.9971	0.9779	0.9874	0.9779	0.984
Upper 95% CI of quad. mean	0.9976	0.9804	0.9892	0.9804	0.9864
Lower 95% CI of quad. mean	0.9966	0.9754	0.9855	0.9754	0.9816
Kurtosis	-1.253	-1.196	3.463	-1.196	-1.391
Skewness	0.6458	0.6007	-1.177	0.6007	0.7904
Sum	18.94	18.58	18.76	18.58	18.7

TABLE 11. One-way analysis of variance (ANOVA) test the proposed AMDTO+RF algorithm.

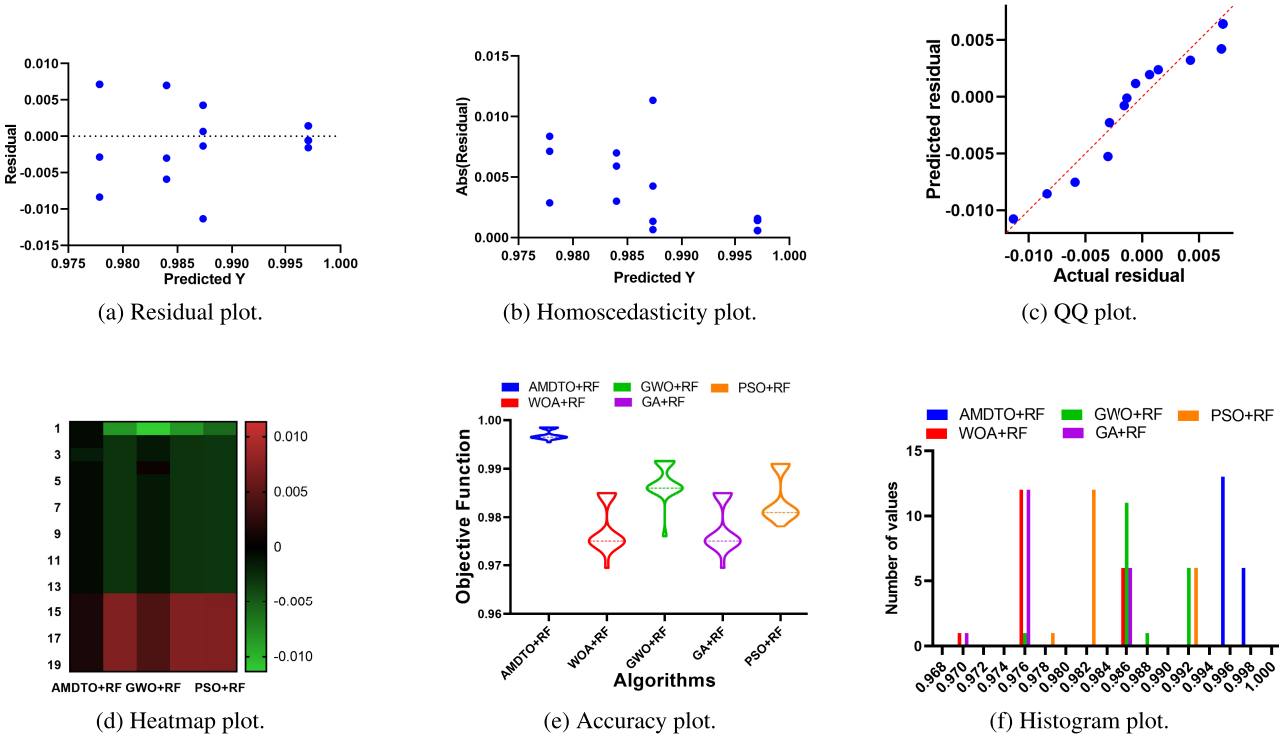
	SS	DF	MS	F(DFn, DFd)	P-value
Treatment (between columns)	0.004826	4	0.001206	F (4, 90) = 65.37	P<0.0001
Residual (within columns)	0.001661	90	0.00001846		
Total	0.006487	94			

of the AMDTO and other four optimizers, namely, WOA, GWO, and PSO. All the optimizers are used to optimize the parameters of RF and the results are recorded and analyzed.

Table 10 presents a statistical analysis of the results achieved by the RF classifier when optimized using the proposed algorithm and using the other optimizers. As shown in the

TABLE 12. Wilcoxon signed-rank test of the results achieved by the optimized RF using five optimization methods.

	AMDTO+RF	WOA+RF	GWO+RF	GA+RF	PSO+RF
Theoretical median	0	0	0	0	0
Actual median	0.9965	0.975	0.986	0.975	0.981
Number of values	19	19	19	19	19
Wilcoxon Signed Rank Test					
Sum of signed ranks (W)	190	190	190	190	190
Sum of positive ranks	190	190	190	190	190
Sum of negative ranks	0	0	0	0	0
P value (two tailed)	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Exact or estimate?	Exact	Exact	Exact	Exact	Exact
P value summary	****	****	****	****	****
Significant (alpha=0.05)?	Yes	Yes	Yes	Yes	Yes
How big is the discrepancy?					
Discrepancy	0.9965	0.975	0.986	0.975	0.981



(a) Residual plot. (b) Homoscedasticity plot. (c) QQ plot.

(d) Heatmap plot. (e) Accuracy plot. (f) Histogram plot.

FIGURE 15. Analysis of the results achieved by the proposed optimized random forest algorithm (AMDTO+RF).

table, the results achieved using the proposed optimization algorithm are better than those achieved by the other algorithms in all statistical criteria included in this analysis. This analysis is performed in terms of the achieved accuracy.

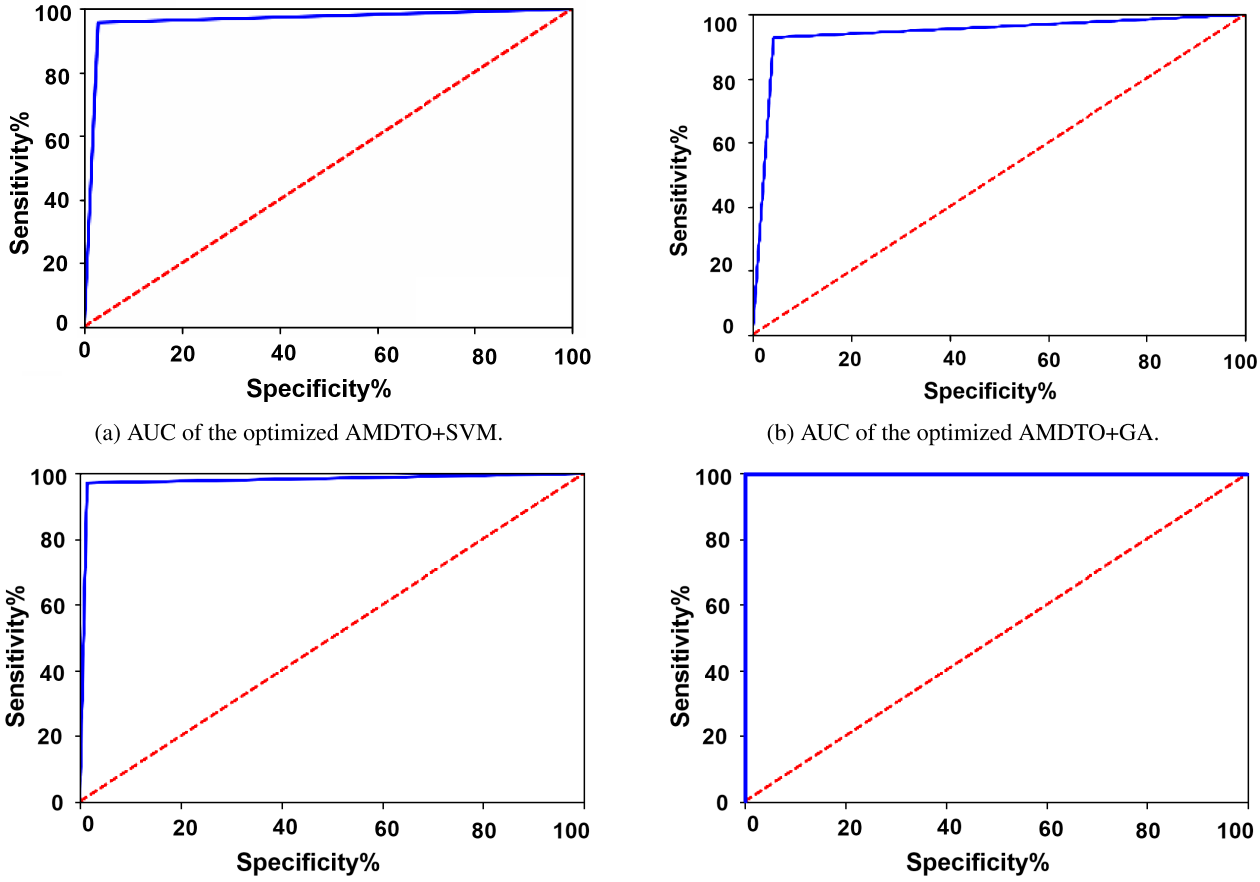
On the other hand, the one-way analysis of variance (ANOVA) test is performed to show the effectiveness of the proposed approach. Table 11 presents the results of this test. As shown in the table, the difference in the mean values is less than (0.0001), which statistically means the efficiency of the proposed approach.

In addition, the statistical Wilcoxon test is performed to measure the difference between the proposed approach and the other approaches based on the other four optimization methods. Table 12 presents the test results. In this table, the results show the significance of all optimizers in classifying the pothole and plain roads. In addition, it can be noted that the discrepancy value using the proposed approach is (0.9965), which is better than those achieved by the other optimizers. These results confirm the effectiveness and superiority of the proposed approach.

Moreover, an analysis of the achieved results using RF classifier based on the proposed optimization algorithm is depicted in Fig. 15. In this figure, six plots are shown. Three plots in the top level, depicts the analyses of the achieved results using the proposed approach. Whereas the three plots in the bottom level compare the results of the proposed approach to the results achieved by the other four optimization methods. Based on these results, it can be clearly shown

TABLE 13. Performance metrics of the achieved results.

Metric	AMDTO+MLP	AMDTO+GAP	AMDTO+SVM	Proposed AMDTO+RF
Accuracy	0.978571428	0.94285714	0.96428571	0.997950637
TPR	0.971428571	0.92857142	0.95714285	0.997506234
TNR	0.985714285	0.95714285	0.97142857	0.998003992
PPV	0.985507246	0.95588235	0.97101449	0.983606557
NPV	0.971830985	0.93055555	0.95774647	0.999700091
F1-score	0.978417266	0.94202898	0.96402877	0.990507635



(a) AUC of the optimized AMDTO+SVM.

(b) AUC of the optimized AMDTO+GA.

(c) AUC of the optimized AMDTO+MLP.

(d) AUC of the proposed AMDTO+RF.

FIGURE 16. AUC of the achieved results based on the optimization of four classifiers using the proposed AMDTO algorithm.

that the proposed approach achieves the best performance among the other optimizers.

G. COMPARISON WITH OTHER MODELS

The last set of conducted experiments targets comparing the proposed optimizer with three classifiers, namely multi-layer perceptron (MLP), global average pooling (GAP), and support vector machines (SVM), in addition to the adopted classifier, RF. The proposed optimization algorithm, AMDTO, is used to optimize the parameters of these classifiers and the classification results are recorded and analyzed. Table 13 presents the evaluation of the achieved results. As shown in the table, the proposed approach, AMDTO+RF, achieved accuracy (99.7950637%), TPR (99.7506234%), TNR (99.8003992%), PPV (98.3606557%), NPV (99.9700091%), F1-score (99.0507635%). These results are better than those achieved by the other optimized models, which confirms the superiority of the proposed approach. On the other hand, the area under curve (AUC) plots are shown in Fig. 16. From these plots, it can be easily noted that the proposed approach can robustly differentiate between the pothole and plan road images.

VI. CONCLUSION AND FUTURE WORK

In this paper, we proposed a new approach for classifying potholes and plain roads. The proposed approach is based on employing the deep network ResNet-50 for extracting high-level features from the input image. In addition, the

significant features are selected using the binary dipper throated optimization algorithm. On the other hand, the dataset is balanced using a proposed optimized SMOTE algorithm. Moreover, the random forest classifier is employed for classifying the selected features. This classifier is optimized using the continuous dipper throated optimization algorithm to achieve the best performance. To prove the superiority of the proposed approach, several experiments were conducted to compare the proposed approach to other optimization methods and three classifiers. In addition, a statistical analysis is performed to assess the stability and efficiency of the proposed approach. The results emphasized the effectiveness and superiority of the proposed approach.

ACKNOWLEDGMENT

We deeply acknowledge Princess Nourah bint Abdulrahman University for supporting this research through Princess Nourah bint Abdulrahman University supporting Project number (PNURSP2022R308), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

REFERENCES

[1] C. Wu, Z. Wang, S. Hu, J. Lepine, X. Na, D. Ainalis, and M. Stettler, "An automated machine-learning approach for road pothole detection using smartphone sensor data," *Sensors*, vol. 20, no. 19, p. 5564, Sep. 2020.

[2] B. Varona, A. Monteserin, and A. Teyseyre, "A deep learning approach to automatic road surface monitoring and pothole detection," *Pers. Ubiquitous Comput.*, vol. 24, no. 4, pp. 519–534, Aug. 2020.

[3] A. Basavaraju, J. Du, F. Zhou, and J. Ji, "A machine learning approach to road surface anomaly assessment using smartphone sensors," *IEEE Sensors J.*, vol. 20, no. 5, pp. 2635–2647, Mar. 2020.

[4] X. Li, D. Huo, D. W. Goldberg, T. Chu, Z. Yin, and T. Hammond, "Embracing crowdsensing: An enhanced mobile sensing solution for road anomaly detection," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 9, p. 412, Sep. 2019.

[5] R. S. Rodrigues, M. Pasin, A. Kozakevicius, and V. Monego, "Pothole detection in asphalt: An automated approach to threshold computation based on the Haar wavelet transform," in *Proc. IEEE 43rd Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, Jul. 2019, pp. 306–315.

[6] H. Bello-Salau, A. M. Aibinu, A. J. Onumanyi, E. N. Onwuka, J. J. Dukiya, and H. Ohize, "New road anomaly detection and characterization algorithm for autonomous vehicles," *Appl. Comput. Informat.*, vol. 16, no. 1/2, pp. 223–239, May 2018.

[7] R. Du, G. Qiu, K. Gao, L. Hu, and L. Liu, "Abnormal road surface recognition based on smartphone acceleration sensor," *Sensors*, vol. 20, no. 2, p. 451, Jan. 2020.

[8] Z. Zheng, M. Zhou, Y. Chen, M. Huo, and D. Chen, "Enabling real-time road anomaly detection via mobile edge computing," *Int. J. Distrib. Sensor Netw.*, vol. 15, no. 11, Nov. 2019, Art. no. 1550147719891319.

[9] I. Moazzam, K. Kamal, S. Mathavan, S. Usman, and M. Rahman, "Metrology and visualization of potholes using the Microsoft Kinect sensor," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 1284–1291.

[10] M. Staniek, "Stereo vision method application to road inspection," *Baltic J. Road Bridge Eng.*, vol. 12, no. 1, pp. 38–47, Mar. 2017.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[12] Z. Zhang, X. Ai, C. K. Chan, and N. Dahnoun, "An efficient algorithm for pothole detection using stereo vision," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 564–568.

[13] C. Koch and I. Brilakis, "Pothole detection in asphalt pavement images," *Adv. Eng. Inform.*, vol. 25, no. 3, pp. 507–515, 2011.

[14] M. Muslim, D. Sulistyaningrum, and B. Setiyono, "Detection and counting potholes using morphological method from road video," in *Proc. AIP Conf.*, vol. 2242, Jun. 2020, Art. no. 030011.

[15] M. H. Yousaf, K. Azhar, F. Murtaza, and F. Hussain, "Visual analysis of asphalt pavement for detection and localization of potholes," *Adv. Eng. Informat.*, vol. 38, pp. 527–537, Oct. 2018.

[16] N.-D. Hoang, "An artificial intelligence method for asphalt pavement pothole detection using least squares support vector machine and neural network with steerable filter-based feature extraction," *Adv. Civil Eng.*, vol. 2018, pp. 1–12, Jun. 2018.

[17] M. B. S. G. Naik and V. Nirmalrani, "Detecting potholes using image processing techniques and real-world footage," in *Cognitive Informatics and Soft Computing*, P. K. Mallick, A. K. Bhoi, G. Marques, and V. Hugo C. de Albuquerque, Eds. Singapore: Springer, 2021, pp. 893–902.

[18] J. Dharneeshkar, S. A. Aniruthan, R. Karthika, and L. Parameswaran, "Deep learning based detection of potholes in Indian roads using YOLO," in *Proc. Int. Conf. Inventive Comput. Technol. (ICICT)*, Feb. 2020, pp. 381–385.

[19] A. Kumar, Chakrapani, D. J. Kalita, and V. P. Singh, "A modern pothole detection technique using deep learning," in *Proc. 2nd Int. Conf. Data, Eng. Appl. (IDEA)*, Feb. 2020, pp. 1–5.

[20] S. Silvister, D. Komandur, S. Kokate, A. Khochare, U. More, V. Musale, and A. Joshi, "Deep learning approach to detect potholes in real-time using smartphone," in *Proc. IEEE Pune Sect. Int. Conf. (PuneCon)*, Dec. 2019, pp. 1–4.

[21] A. Ibrahim, S. Mirjalili, M. El-Said, S. S. M. Ghoneim, M. M. Al-Harthi, T. F. Ibrahim, and E.-S.-M. El-Kenawy, "Wind speed ensemble forecasting based on deep learning using adaptive dynamic optimization algorithm," *IEEE Access*, vol. 9, pp. 125787–125804, 2021.

[22] E.-S. M. El-kenawy, S. Mirjalili, S. S. M. Ghoneim, M. M. Eid, M. El-Said, Z. S. Khan, and A. Ibrahim, "Advanced ensemble model for solar radiation forecasting using sine cosine algorithm and Newton's laws," *IEEE Access*, vol. 9, pp. 115750–115765, 2021.

[23] H. Song, K. Baek, and Y. Byun, "Pothole detection using machine learning," in *Advanced Science and Technology Letters*, vol. 150. Science & Engineering Research Support Society, 2018, pp. 151–155, doi: 10.14257/astl.2018.150.35.

[24] E.-S. M. El-kenawy, A. Ibrahim, S. Mirjalili, M. M. Eid, and S. E. Hussein, "Novel feature selection and voting classifier algorithms for COVID-19 classification in CT images," *IEEE Access*, vol. 8, pp. 179317–179335, 2020.

[25] E.-S.-M. El-kenawy, A. Ibrahim, N. Bailek, K. Bouchouicha, M. A. Hassan, M. Jamei, and N. Al-Ansari, "Sunshine duration measurements and predictions in Saharan Algeria region: An improved ensemble learning approach," *Theor. Appl. Climatol.*, vol. 147, nos. 3–4, pp. 1015–1031, Nov. 2021.

[26] A. Ibrahim, H. A. Ali, M. M. Eid, and E.-S.-M. El-kenawy, "Chaotic Harris hawks optimization for unconstrained function optimization," in *Proc. 16th Int. Comput. Eng. Conf. (ICENCO)*, Dec. 2020, pp. 153–158.

[27] M. M. Eid, E.-S.-M. El-kenawy, and A. Ibrahim, "A binary sine cosine-modified whale optimization algorithm for feature selection," in *Proc. Nat. Comput. Colleges Conf. (NCCC)*, Mar. 2021, pp. 1–6.

[28] S. S. M. Ghoneim, T. A. Farrag, A. A. Rashed, E.-S.-M. El-Kenawy, and A. Ibrahim, "Adaptive dynamic meta-heuristics for feature selection and classification in diagnostic accuracy of transformer faults," *IEEE Access*, vol. 9, pp. 78324–78340, 2021.

[29] E.-S.-M. El-Kenawy, S. Mirjalili, F. Alassery, Y.-D. Zhang, M. M. Eid, S. Y. El-Mashad, B. A. Aloyaydi, A. Ibrahim, and A. A. Abdelhamid, "Novel meta-heuristic algorithm for feature selection, unconstrained functions and engineering problems," *IEEE Access*, vol. 10, pp. 40536–40555, 2022.

[30] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama, and H. Omata, "Road damage detection and classification using deep neural networks with smartphone images," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 12, pp. 1127–1141, Jun. 2018.

[31] A. Rasyid, F. Yudianto, H. Wicaksono, M. R. U. Albaab, M. F. Falah, Y. Y. F. Panduman, A. A. Yusuf, D. K. Basuki, A. Tjahjono, R. P. N. Budiarti, and S. Sukaridhoto, "Pothole visual detection using machine learning method integrated with Internet of Thing video streaming platform," in *Proc. Int. Electron. Symp. (IES)*, Sep. 2019, pp. 672–675.

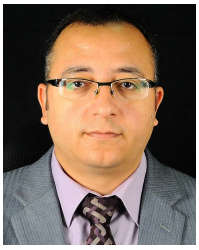
[32] U. Bhatt, S. Mani, E. Xi, and J. Z. Kolter, "Intelligent pothole detection and road condition assessment," in *Proc. Bloomberg Data Good Exchange Conf.*, Chicago, IL, USA, vol. abs/1710.02595, 2017, doi: 10.48550/arXiv.1710.02595.

[33] H.-W. Wang, C.-H. Chen, D.-Y. Cheng, C.-H. Lin, and C.-C. Lo, "A real-time pothole detection approach for intelligent transportation system," *Math. Problems Eng.*, vol. 2015, pp. 1–7, Aug. 2015.
[34] A. Mednis, G. Strazdins, R. Zviedris, G. Kanonirs, and L. Selavo, "Real time pothole detection using Android smartphones with accelerometers," in *Proc. Int. Conf. Distrib. Comput. Sensor Syst. Workshops (DCOSS)*. Barcelona, Spain: Casa Convalescència, 2011, pp. 1–6.
[35] K. Pawar, S. Jagtap, and S. Bhoir, "Efficient pothole detection using smartphone sensors," in *Proc. ITM Web Conf.*, vol. 32, 2020, p. 3013.
[36] A. S. El-Wakeel, J. Li, A. Noureldin, H. S. Hassanein, and N. Zorba, "Towards a practical crowdsensing system for road surface conditions monitoring," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4672–4685, Dec. 2018.
[37] R. Bustamante-Bello, A. García-Barba, L. A. Arce-Saenz, L. A. Curiel-Ramirez, J. Izquierdo-Reyes, and R. A. Ramirez-Mendoza, "Visualizing street pavement anomalies through fog computing V2I networks and machine learning," *Sensors*, vol. 22, no. 2, p. 456, Jan. 2022.
[38] A. Angulo, J. A. V.-Fernádez, L. M. Aguilar-Lobo, S. Natraj, and G. Ochoa-Ruiz, "Road damage detection acquisition system based on deep neural networks for physical asset management," in *Advances in Soft Computing*. Cham, Switzerland: Springer, 2019, pp. 3–14, doi: 10.1007/978-3-030-33749-0_1.
[39] E. N. Ukhwah, E. M. Yuniarno, and Y. K. Suprapto, "Asphalt pavement pothole detection using deep learning method based on Yolo neural network," in *Proc. Int. Seminar Intell. Technol. Its Appl. (ISITIA)*, Aug. 2019, pp. 35–40.
[40] H. Chen, M. Yao, and Q. Gu, "Pothole detection using location-aware convolutional neural networks," *Int. J. Mach. Learn. Cybern.*, vol. 11, no. 4, pp. 899–911, Apr. 2020.
[41] Aparna, Y. Bhatia, R. Rai, V. Gupta, N. Aggarwal, and A. Akula, "Convolutional neural networks based potholes detection using thermal imaging," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 3, pp. 578–588, Mar. 2022.
[42] A. Kumar, Chakrapani, D. J. Kalita, and V. P. Singh, "A modern pothole detection technique using deep learning," in *Proc. 2nd Int. Conf. Data, Eng. Appl. (IDEA)*, Feb. 2020, pp. 1–5.
[43] M. Muslim, D. Sulistyaningrum, and B. Setiyono, "Detection and counting potholes using morphological method from road video," in *Proc. AIP Conf.*, vol. 2242, Jun. 2020, Art. no. 030011.
[44] N.-D. Hoang, T.-C. Huynh, and V.-D. Tran, "Computer vision-based patched and unpatched pothole classification using machine learning approach optimized by forensic-based investigation metaheuristic," *Complexity*, vol. 2021, pp. 1–17, Sep. 2021.
[45] A. E. Takieldeen, E.-S. M. El-kenawy, M. Hadwan, and R. M. Zaki, "Dipper throated optimization algorithm for unconstrained function and feature selection," *Comput., Mater. Continua*, vol. 72, no. 1, pp. 1465–1481, 2022.
[46] R. Bello, Y. Gomez, A. Nowe, and M. M. Garcia, "Two-step particle swarm optimization to solve the feature selection problem," in *Proc. 7th Int. Conf. Intell. Syst. Design Appl. (ISDA)*, Oct. 2007, pp. 691–696.
[47] M. M. Kabir, M. Shahjahan, and K. Murase, "A new local search based hybrid genetic algorithm for feature selection," *Neurocomputing*, vol. 74, no. 17, pp. 2914–2928, 2011.
[48] H. Xia, Y. M. Akay, and M. Akay, "Selecting relevant genes from microarray datasets using a random forest model," *IEEE Access*, vol. 9, pp. 97813–97821, 2021.
[49] K. Kim, "Noise avoidance SMOTE in ensemble learning for imbalanced data," *IEEE Access*, vol. 9, pp. 143250–143265, 2021.
[50] K. T. Chui, R. W. Liu, M. Zhao, and P. O. D. Pablos, "Predicting students' performance with school and family tutoring using generative adversarial network-based deep support vector machine," *IEEE Access*, vol. 8, pp. 86745–86752, 2020.
[51] E. M. Hassib, A. I. El-Desouky, E.-S.-M. El-Kenawy, and S. M. El-Ghamrawy, "An imbalanced big data mining framework for improving optimization algorithms performance," *IEEE Access*, vol. 7, pp. 170774–170795, 2019.
[52] R. Rashid, S. Ubaid, M. Idrees, R. Rafi, and I. S. Bajwa, "Visualization of salient object with saliency maps using residual neural networks," *IEEE Access*, vol. 9, pp. 104626–104635, 2021.
[53] A. Kumar, "Pothole detection dataset," Apr. 21, 2022. [Online]. Available: https://www.kaggle.com/atulyakumar98/pothole-detection-dataset
[54] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, "Advanced deep-learning techniques for salient and category-specific object detection: A survey," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 84–100, Jan. 2018.
[55] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.
[56] A. Al-Dhamari, R. Sudirman, and N. H. Mahmood, "Transfer deep learning along with binary support vector machine for abnormal behavior detection," *IEEE Access*, vol. 8, pp. 61085–61095, 2020.
[57] S. Yu, L. Xie, L. Liu, and D. Xia, "Learning long-term temporal features with deep neural networks for human action recognition," *IEEE Access*, vol. 8, pp. 1840–1850, 2020.
[58] E.-S. M. El-Kenawy and M. Eid, "Hybrid gray wolf and particle swarm optimization for feature selection," *Int. J. Innov. Comput. Inf. Control*, vol. 16, no. 3, pp. 831–844, 2020.
[59] F. A. Şenel, F. Gökçe, A. S. Yuksel, and T. Yigit, "A novel hybrid PSO–GWO algorithm for optimization problems," *Eng. Comput.*, vol. 35, no. 4, pp. 1359–1373, Dec. 2019.
[60] E.-S. M. El-Kenawy, M. M. Eid, M. Saber, and A. Ibrahim, "MbGWO-SFS: Modified binary grey wolf optimizer based on stochastic fractal search for feature selection," *IEEE Access*, vol. 8, pp. 107635–107649, 2020.
[61] S. Mugemanyi, Z. Qu, F. X. Rugema, Y. Dong, C. Bananeza, and L. Wang, "Optimal reactive power dispatch using chaotic bat algorithm," *IEEE Access*, vol. 8, pp. 65830–65867, 2020.
[62] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Adv. Eng. Softw.*, vol. 95, pp. 51–67, Feb. 2016.
[63] X. Zhang, D. Wang, and H. Chen, "Improved biogeography-based optimization algorithm and its application to clustering optimization and medical image segmentation," *IEEE Access*, vol. 7, pp. 28810–28825, 2019.
[64] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, "Multi-verse optimizer: A nature-inspired algorithm for global optimization," *Neural Comput. Appl.*, vol. 27, no. 2, pp. 495–513, Feb. 2016.
[65] S. H. S. Moosavi and V. K. Bardsiri, "Satin bowerbird optimizer: A new optimization algorithm to optimize anfis for software development effort estimation," *Eng. Appl. Artif. Intell.*, vol. 60, pp. 1–15, Apr. 2017.
[66] I. Fister, X.-S. Yang, I. Fister, and J. Brest, "Memetic firefly algorithm for combinatorial optimization," in *Bioinspired Optimization Methods and Their Applications*, Bohinj, Slovenia, 2012, pp. 75–86, doi: 10.48550/arXiv.1204.5165.

AMEL ALI ALHUSSAN received the B.Sc., M.Sc., and Ph.D. degrees in computer and information sciences from King Saud University, Saudi Arabia. Her M.Sc. thesis in software engineering and her Ph.D. thesis in artificial intelligent. She is currently an Assistant Professor with the Computer Sciences Department, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University (PNU), Saudi Arabia. She is working in her college in various administrative and academic positions. Her research interests include machine leaning, networking, and software engineering.

DOAA SAMI KHAFAGA received the B.Sc. degree (Hons.) in computer and information sciences in the field of computer science and the M.Sc. and Ph.D. degrees in computer science from the College of Faculty of Computers and Artificial Intelligence, Helwan University, Egypt, in 2003, 2008, and 2013, respectively. She has 18 years academic experience, as she has worked at the Computer Science Department, College of Information Technology and Artificial Intelligence, Misr University for Science and Technology, Egypt; the Computer Science Department, Institute of Public Administration, Saudi Arabia; and the Computer Science Department, Faculty of Computer and Information Sciences, Princess Nourah bint Abdulrahman University (PNU). Her main research interests include data science, artificial intelligence, machine learning, data mining, and software engineering. She also has FHEA, the Fellow Recognition from U.K. Higher Education Academy. She is currently a reviewer in some journals.

EL-SAYED M. EL-KENAWY (Senior Member, IEEE) is currently an Assistant Professor at the Delta Higher Institute for Engineering and Technology (DHIET), Mansoura, Egypt. He has published more than 35 papers with more than 1300 citations and an H-index of 24. He has launched and pioneered independent research programs. He motivates and inspires his students in different ways by providing a thorough understanding of various computer concepts. He explains complex concepts in an easy-to-understand manner. His research interests include artificial intelligence, machine learning, optimization, deep learning, digital marketing, and data science. He is a Reviewer for *Computers, Materials and Continua* journal, IEEE Access, and other journals.



ABDELHAMEED IBRAHIM (Member, IEEE) received the bachelor's and master's degrees in engineering from the Department of Computer Engineering and Systems, in 2001 and 2005, respectively, and the Ph.D. degree in engineering from the Faculty of Engineering, Chiba University, Japan, in 2011. He was with the Faculty of Engineering, Mansoura University, Egypt, from 2001 to 2007, where he is currently an Associate Professor in computer engineering. He has published more than 60 publications with over 1800 citations and an H-index of 22. His research interests include machine learning, optimization, swarm intelligence, and pattern recognition. He serves as a Reviewer (75 verified reviews based on publons) for IEEE Access, *Computer Standards and Interfaces*, *Optical Engineering*, IEEE Journal of Biomedical and Health Informatics, *Biomedical Signal Processing and Control*, *IET Image Processing*, *Multimedia Tools and Applications*, *Frontiers of Information Technology and Electronic Engineering*, *Computers, Materials and Continua*, *Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization*, *Journal of Healthcare Engineering*, *Sensors*, *Materials*, *Applied Sciences*, *Entropy*, and *Healthcare* journals.

MARWA METWALLY EID (Member, IEEE) received the Ph.D. degree in electronics and communications engineering from the Faculty of Engineering, Mansoura University, Egypt, in 2015. She has been working as an Assistant Professor at the Delta Higher Institute for Engineering and Technology, since 2011. Her current research interests include image processing, encryption, wireless communication systems, and field programmable gate array (FPGA) applications.

ABDELAZIZ A. ABDELHAMID received the M.Sc. degree in computer science from the Faculty of Computer and Information Sciences, Ain Shams University, and the Ph.D. degree in computer engineering from the Faculty of Engineering, Auckland University, New Zealand. He is currently an Assistant Professor with the Department of Computer Science, Faculty of Computer and Information Sciences, Ain Shams University. He is also working as an Assistant Professor with the Computer Science Department, College of Computing and Information Technology, Shaqra University. His research interests include speech and image processing and machine learning-based intelligent systems.