

Received April 2, 2020, accepted April 27, 2020, date of publication May 6, 2020, date of current version May 18, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2991968

Using Hybrid Filter-Wrapper Feature Selection With Multi-Objective Improved-Salp Optimization for Crack Severity Recognition

ESRAA ELHARIRI¹, NASHWA EL-BENDARY², (Senior Member, IEEE),
AND SHEREEN A. TAIE¹

¹Department of Computer Science, Faculty of Computers and Information, Fayoum University, Fayoum 63514, Egypt

²Arab Academy for Science, Technology and Maritime Transport (AASTMT), Giza B 2401, Egypt

Corresponding author: Esraa Elhariri (emh00@fayoum.edu.eg)

ABSTRACT The emerging technology of Structural Health Monitoring (SHM) paved the way for spotting and continuous tracking of structural damage. One of the major defects in historical structures is cracking, which represents an indicator of potential structural deterioration according to its severity. This paper presents a novel crack severity recognition system using a hybrid filter-wrapper with multi-objective optimization feature selection method. The proposed approach comprises two main components, namely, (1) feature extraction based on hand-crafted feature engineering and CNN-based deep feature learning and (2) feature selection using hybrid filter-wrapper with a multi-objective improved salp swarm optimization. The proposed approach is trained and validated by utilizing 10 representative UCI datasets and 4 datasets of crack images. The obtained experimental results show that the proposed system enhances the performance of crack severity recognition with $\approx 37\%$ and $\approx 31\%$ increase in recognition average accuracy and F-measure, respectively. Also, a reduction rate of $\approx 67\%$ is achieved in the extracted feature set with all the tested datasets compared to the conventional classification approaches using the whole set of features. Moreover, the proposed approach outperforms other approaches with classical feature selection methods in terms of feature reduction rate and computational time. It is noticed that using VGG16 learned features outperforms using the fused hand-crafted features by 17.7%, 15.9%, and 23.5% for fine, moderate, and severe crack recognition, respectively. The significance of this paper is to investigate and highlight the impact of applying multi-feature dimensionality reduction through adopting hybrid filter-wrapper with multi-objective optimization methods for feature selection considering the case study of crack severity recognition for SHM.

INDEX TERMS Crack detection, feature selection, Fisher score, hybrid filter-wrapper, Kappa index, multi-objective optimization, salp optimization algorithm, severity recognition, structural health monitoring.

I. INTRODUCTION

Protection and preservation of historical buildings have attracted the interest of many researchers due to their importance. There are many threats facing historical buildings such as environmental and human factors inducing stresses inside the buildings and resulting in the degradation process over time [1]–[3]. Therefore, developing automated systems for detecting historical building damage is a pivotal step to save cultural heritage.

Structural Health Monitoring (SHM) is considered as a substantial method for damage detection and estimation of

structures durability by using multidisciplinary fields such as sensors, image processing, materials, system integration, signal processing, and interpretation. The main aim of SHM technology is not solely detecting structural failures/deterioration, but also providing an early indication of materialistic damage. Consequently, those provided early warnings can be used to determine repair strategies before the structural damage results in failure [4], [5].

Cracks represent major defect in building infrastructure, with a critical economic impact and can constitute safety risks if left without care. Also, cracks are considered the main indicators of potential structural damage and building's durability. In order to help safeguard historical buildings from further damage and to understand future development of

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaou Li².

structural defects, it is necessary to develop various effective and adaptable monitoring strategies to assess the state of damaged structures/buildings. Observation results could be used to plan and optimize maintenance and conservation works. Usually, cracks are distinguished by dark areas in an image. This makes it easy for a human to identify cracks, however, presents difficulties for human judgment to classify crack severity in addition to time consumption, and error-prone. Hence, the automation of crack detection and crack severity recognition processes is a challenging task to obtain information about building's structural health conditions [6], [7].

Nowadays, many researchers widely use several Machine Learning (ML) and Deep Learning (DL) techniques for crack detection. The performance of these techniques is strongly influenced by a variety of factors such as the size of training data, number of features, algorithm parameters, and in some cases, the nature and complexity of the studied problem. Whereas, feeding ML techniques with a huge number of features often causes several challenges such as high computation complexity, overfitting, and low interpretability of the final trained model. This is due to the curse of dimensionality arising from the proportion between the number of features and the number of samples [8]–[10]. Hence, to improve the performance of crack detection systems, there is a need to resolve this problem. Reducing the number of features is the most common way to handle the dimensionality problem using either feature selection or feature construction.

Feature selection (FS) is one of the most substantial preprocessing steps in any classification task. The main objective of FS is to pick out the best subset of features via removing extraneous or redundant features to effectively reduce the dimensionality of data, shorten the training time, and enhance the classification accuracy. Utilizing the brute-force method to achieve FS leads to a high computational cost, with considerable risk of overfitting in most circumstances. On the other hand, the manual selection of prevalent features is inconsistent/infeasible in most cases. Hence, searching for the optimal feature subset, which accurately detects cracks, is a highly demanding task [11]–[13].

Feature selection methods can be generally classified into three types, namely, wrapper-based, filter-based, and hybrid methods, which are differed in the evaluation approach. Wrapper-based methods employ a ML algorithm to compute the quality of the selected subset of features. Although, it obtains better accuracy than the filter-based methods, it has the following drawbacks: 1) the selected features are dependent on the learner and may be not optimal for other learners, 2) the high computational cost, 3) parameters fine-tuning of the learners may be time consuming, and 4) the inherent limitations of the learners. In contrast to wrapper-based methods, filter-based methods use predefined metrics or statistical measures (e.g. mutual information) to evaluate the selected features. Even though, these approaches are fast and independent of the learners, they have the drawbacks of lacking the interaction between learners and features

in addition to, lacking the ability to work with redundant features [11]–[13].

In this work, we seek to integrate the power of the two illustrated feature selection processes through developing a multi-objective filter-wrapper feature selection method, which will be utilized within the framework of a multi-objective classification approach.

Moreover, during the last two decades, many state-of-the-art meta-heuristic algorithms such as Particle Swarm Optimization (PSO), Cuckoo-Search Algorithm (CSA), Artificial Bee Colony (ABC), Whale Optimization Algorithm (WOA), Genetic Algorithm (GA), Firefly Algorithm (FA), Bat Algorithm (BA), Grey Wolf Optimization (GWO), Salp Swarm Algorithm (SSA) etc. are utilized to solve the problem of feature selection due to their efficient and powerful performance in handling several complex real-world problems [11]–[13].

Accordingly, in this paper, to improve the performance of crack detection and severity recognition, we present a multi-objective filter-wrapper feature selection method, based on an improved Binary Salp Swarm Algorithm (BSSA). The SSA mimics the attitude of salps during the foraging and navigation in oceans to conduct global optimization [14].

The significance of this paper is to improve the performance of crack detection and severity recognition through the optimal fusion/selection of hand-crafted features such as Local Binary Patterns (LBP) and Histogram of Oriented Gradients (HOG) and Convolutional Neural Networks (CNN) based learned features.

The main contributions of this paper are summarized in the following points:

- Constructing expert-based annotated primary datasets of crack images, collected over 2 years from a historical location.
- Developing a multi-objective improved BSSA (MO-IBSSA) approach with Kappa index as a fitness function for handling the feature selection problem.
- Designing and investigating the use of feature fusion of LBP and HOG hand-crafted features and CNN-based learned features with MO-IBSSA optimization.
- Establishing a novel hybrid filter-wrapper feature selection method based on MO-IBSSA for crack detection crack severity recognition.
- Testing and Validating the performance of the proposed approach through implementing multiple experiments on the primary obtained real-world datasets and other selected publicly available benchmark datasets.

The remaining of this paper is organized as follows. Section II presents state-of-the-art studies related to feature selection problem and crack detection. An overview of the methods used in the proposed approach is explained in section III. Phases of the proposed crack severity recognition system are described in sections IV. Section V presents and discusses the obtained experimental results. Conclusions and future work are introduced in section VI.

II. RELATED WORK

Recently, feature selection, fusion, as well as structural damage detection have become very important issues. As mentioned in the previous section, there are many studies tackling these problems that were conducted using diverse ML and DL methods. This section provides a review of state-of-the-art literature that addressed feature selection and fusion techniques besides crack detection approaches.

A. FEATURE SELECTION/FUSION METHODS

Recently, various studies addressed several feature selection and fusion methods for achieving considerable advantages for classification performance.

In [15], Kaur *et al.* presented a hybrid feature weighting approach for enhancing the diagnostic accuracy of Magnetic Reasoning (MR) brain tumor classification. The presented approach integrated Fisher Criterion and Parameter-Free BAT optimization (PFree BAT) algorithm for weighting conventional metabolite ratio features extracted from short echo MR spectroscopy. The K-nearest neighbor (KNN) algorithm with 5-fold cross-validation achieved an accuracy, sensitivity, and specificity of 93.72%, 96%, and 91%, respectively.

Sasikala *et al.* proposed a feature fusion approach based on FA with Optimum-Path Forest classifier fitness function [16] for enhancing breast cancer detection. The proposed approach serially fused LBP features extracted from both Craniocaudal (CC) and Mediolateral Oblique (MLO) view mammogram images. The Support Vector Machine (SVM) classifier achieved detection accuracy of 96.6% and F-measure of 96.53%, respectively, using DDSM dataset. Moreover, the same study achieved detection accuracy of 86.5% and F-measure of 82.88%, respectively, using INbreast dataset.

Also, Gunasundari *et al.* in [17] proposed a wrapper-based feature selection approach, based on Boolean Particle Swarm Optimization (BoPSO), for improving classification accuracy in liver and kidney disease diagnosis. The proposed approach used first-order and second-order statistical features extracted from Grey Level Co-Occurrence Matrix (GLCM) using abdominal CT image slices. The proposed approach presented two new modified BPSO algorithms, namely, Velocity Bounded BoPSO (VbBoPSO) and Improved Velocity Bounded BoPSO (IVbBoPSO) with Probabilistic Neural Network (PNN)/SVM classifier fitness function. The PNN/SVM achieved accuracies of 77.14% and 82.86%, respectively, for liver diseases using elite features selected by IVbBoPSO. On the other hand, PNN/SVM achieved accuracies of 77.14% and 90%, respectively, for kidney diseases using elite features selected by VbBoPSO.

Also in [18], Alirezaei *et al.* proposed a wrapper-based method for feature selection using four meta-heuristic algorithms, namely, Non-dominated Sorting Genetic Algorithm II (NSGA-II), FA, PSO, and Imperialist Competitive Algorithm (ICA), with bi-objective fitness for reducing features dimensionality and improving classification accuracy

in diabetes diagnosis. The SVM classifier achieved classification accuracies of 100%, 100%, 98.2%, and 94.6% when evaluating the features selected by FA, ICA, NSGA-II, and PSO, respectively, using the PIMA Indian Type-2 diabetes dataset obtained from UCI Machine Learning Datasets Repository. The depicted results showed that both FA and ICA algorithms outperformed the NSGA-II and PSO.

Moreover, González *et al.* proposed in [19], a wrapper-based method for multi-objective feature selection based on NSGA-II evolutionary algorithm with Linear Discriminate Analysis (LDA) fitness for enhancing the performance of Brain Computer Interfaces (BCI) systems. A new objective function and feature ranking procedure were proposed for analyzing stability of the proposed method. It used the second moment (variance) of the wavelet coefficient set as features. Evaluating the proposed method on the collected EEG signals using Naive Bayesian Classifier (NBC), KNN, LDA, and combinations of these classifiers proved that KNN obtained the best Kappa values on the test dataset.

In [14], Ibrahim *et al.* presented a feature selection approach based on an improved SSA for dimensionality reduction. The presented approach integrated both SSA and PSO as a hybrid optimization technique (SSAPSO) to increase the convergence rate and enhance the effectiveness of SSA for the exploration and exploitation. The proposed SSAPSO approach was evaluated on a set of 15 benchmark functions and proved its performance enhancement without affecting the computational time. Also, feeding the KNN with the features selected by the SSAPSO achieved an accuracy of 74.8%-98.31% and F-measure of 78%-98.34% on evaluating with 10 representative datasets obtained from the UCI repository.

Additionally, in [20], Aljarah *et al.* proposed a feature selection approach based on asynchronous accelerating multi-leader salp chain with KNN and number of features fitness to avoid trapping in local solution when applied to high-dimensional datasets. The proposed approach improved the BSSA via adding 3 different asynchronous updating rules and a novel leadership structure. Through feeding the KNN with the features selected by the proposed approach, an accuracy of 73.64%-99.77% was achieved using 70% of the 20 selected UCI datasets, compared to the other implemented optimization techniques.

Farisa *et al.* proposed two wrapper-based feature selection approaches based on BSSA with KNN and number of features fitness to test the ability of SSA in feature selection problems [21]. In the first approach, the continuous version of conventional SSA was converted to binary using eight transfer functions to be applied for feature selection. While, the second approach replaced the average operator by utilizing the crossover operator with the transfer function to improve the exploratory behavior of BSSA. Through feeding the KNN with the features selected by the proposed BSSA, an accuracy of 73.35%-100% was achieved using around 90% of the 22 selected UCI datasets, compared to the

BGWO, Binary Gravitational Search Algorithms (BGSA), BBA, BPSO, and GA.

Another variant of SSA was proposed in [22] by Hegazy *et al.*, by introducing a new control parameter called inertia weight, used to adjust the present best solution in order to improve solution accuracy, convergence speed, and reliability. The proposed algorithm was applied along with KNN and number of selected features fitness for feature selection problems. The proposed feature selection algorithm was evaluated on a set of 23 selected UCI datasets and proved its superior performance in terms of accuracy and feature reduction rate, against other optimizers. The KNN achieved an accuracy of 63.19%-98.99% when evaluated with 23 datasets obtained from the UCI repository.

Also, in [23], to tackle the problem of slow convergence speed and getting stuck in local optima of SSA, Hegazy *et al.* presented an improved SSA by combining five chaotic maps with SSA (CSSA). The proposed CSSA with KNN and number of selected features fitness was applied for feature selection problems. Through employing the KNN with the features selected by the proposed CSSA, an accuracy of 60.44%-97.83% was achieved using around 92.6% of the 27 selected UCI datasets, compared to the standard SSA and other optimizers, especially, when using the Tent chaos.

Zawbaa *et al.* developed a feature selection approach based on hybrid bio-inspired heuristic [24] for solving the dimensionality reduction problem of large datasets with small instance-set. The developed approach combined the Ant-Lion Optimization (ALO) algorithm with the GWO to form a hybrid bio-inspired optimization algorithm ALO-GWO. The developed ALO-GWO takes the characteristics of diversification and stagnation avoidance from ALO and the characteristic of faster convergence from GWO. Evaluating the ALO-GWO with the KNN as well as the number of selected features fitness function on a set of 18 different datasets from the UCI repository showed that the developed approach outperformed other optimization algorithms such as GA, PSO, GWO, and ALO. Moreover, the proposed approach achieved fitness value of 0-0.192 and 0-0.178, when evaluating on 7 different Micro-array gene expression datasets and 5 different face detection image datasets, respectively.

In a similar vein, Chantar *et al.* [25] presented a wrapper-based method for feature selection based on improved Binary GWO (BGWO) with elite-based crossover for enhancing the classification accuracy of Arabic text. The proposed method employed Term-frequency Inverse Document Frequency (TF-IDF) for term weighting. It presented an improved BGWO with KNN and the number of selected features as a single objective fitness function. Feeding decision trees with the features selected by the proposed BGWO achieved F-measures of 0.8255, 0.73663, and 0.79702, however, using Naive Bayes achieved F-measures of 0.93445, 0.82036, and 0.86686, using SVM achieved F-measures of 0.9616, 0.91706, and 0.90473.

After that, Abdel-Basset *et al.* [26] proposed a new wrapper-based feature selection method based on a combi-

nation of GWO with two-phase mutation to strengthen the exploitation capability of the GWO. Moreover, the effect of two different transformation functions were studied. The two mutation phases aimed to reduce the number of selected features while maintaining high classification accuracy and tried to add more enlightening features to enhance classification accuracy. Feeding the KNN classifier with the features selected by the proposed method achieved an accuracy of 33%-98.88% using around 94.3% of the 35 selected UCI datasets, compared to other algorithms.

In [27], Mafarja *et al.* presented two different binary variants of WOA to solve feature selection problems, where, a novel wrapper-based feature selection method based on WOA with KNN and the number of selected features fitness was proposed to improve classification accuracy. In the first variant, the random selection operator in the searching process was replaced by both Tournament and Roulette Wheel selection mechanisms. While, in the second variant, WOA was equipped with crossover and mutation operators to improve the exploitation capability in the WOA. The performance of the proposed method was evaluated using 20 selected datasets from the UCI repository. The experimental results showed that WOA equipped with crossover and mutation outperformed other algorithms using around 70% of the datasets with an accuracy of 78.5%-100%.

In [28], Thaher *et al.* presented a wrapper-based feature selection method for high-dimensional low sample size datasets, based on binary Harris Hawks optimizer (HHO) with KNN fitness. The S-shaped and V-shaped transfer functions were utilized to convert the continuous HHO algorithm to a binary HHO (BHHO). To evaluate the proposed BHHO algorithm for feature selection, 9 online public high-dimensional datasets with low samples were used. The obtained results showed that BHHO with S-shaped transfer function achieved higher accuracy for 6 datasets, that is, an accuracy of 54.1%-100% was obtained by feeding the KNN classifier with the features selected by the proposed BHHO using around 55.56% of the used datasets.

Another binary variant of HHO, called Quadratic Binary Harris Hawk Optimization (QBHHO), for solving feature selection problem was proposed in [29] by Too *et al.* The proposed QBHHO used four quadratic transfer functions to transform the HHO into a binary one. To validate the proposed approach, 22 datasets selected from the UCI repository were used. Feeding the KNN with the features selected by the proposed QBHHO using the fourth quadratic transfer function achieved an accuracy of 39.78%-97.76% using around 50% of the tested datasets.

In [30], Sindhu *et al.* proposed a wrapper-based feature selection method based on Improved Sine-Cosine Algorithm (ISCA), through combining the SCA with the Elitism strategy and new updating mechanism for the best solution to enhance classification accuracy. The performance of the proposed ISCA was validated with 10 datasets selected from the UCI repository. With feeding Extreme Learning Machine (ELM) classifier with the features selected by the proposed

TABLE 1. Summary of surveyed state-of-the-art studies for meta-heuristic based feature selection/fusion methods (SO: single-objective, MO: multi-objective).

| Study | Objectives | Features | Fitness Function | Optimization Technique | Feature Combination | Classifier | Dataset | Performance Measures |
|----------------|--|---|---|---------------------------------|---------------------|---------------------------------|--|---|
| [16] 2020 | Feature fusion approach for enhancing breast cancer detection | LBP from MLO and CC view mammograms | SO Optimum-Path Forest classifier | BFA | ✓ (fusion) | SVM | DDSM and INbreast datasets | DDSM : Sensitivity: 95% Specificity: 98.1% Accuracy: 96.6% F-measure: 96.53% INbreast: Sensitivity: 72.3% Specificity: 98.2% Accuracy: 86.5% F-measure: 82.88% |
| [26] 2020 | Wrapper-based feature selection approach to reduce the number of selected features while maintaining high classification accuracy | Various types of features | SO KNN and the number of selected features | GWO with two-phase mutation | ✓ (selection) | KNN | 35 UCI datasets | Accuracy ranges between 33% and 98.88% |
| [28] (2020) | Wrapper-based feature selection approach to reduce the number of selected features while maintaining high classification accuracy for high-dimensional, low sample size datasets | Various types of features | SO KNN and the number of selected features | BHBO | ✓ (selection) | KNN | 9 online public high-dimensional, low samples datasets | Accuracy ranges between 54.1% and 100% |
| [29] 2019 | Wrapper-based feature selection approaches to reduce the number of selected features while maintaining high classification accuracy | Various types of features | SO KNN and the number of selected features | QBHBO | ✓ (selection) | KNN | 22 UCI datasets | Accuracy ranges between 39.78% and 97.76% |
| [14] 2019 | A hybrid optimization-based feature selection method to remove redundant or confusing features and improve the efficacy of SSA | Various types of features | SO KNN and the number of selected features | Hybrid SSA-PSO | ✓ (selection) | KNN | 10 UCI datasets | Accuracy ranges between 74.8% and 98.31% F-measure ranges between 78% and 98.34% |
| [19] 2019 | A wrapper-based method for feature selection to enhance the performance of Brain Computer Interfaces (BCI) systems | The second moment (variance) of the wavelet coefficient set | MO Multiple LDA | NSGA-II | ✓ (selection) | KNN, NBC, LDA+KNN, LDA+NBC | EEG signals Motor Imagery (MI) dataset | KNN obtains the best Kappa values (0.59 and 0.704 for two subjects) on the test dataset |
| [25] 2019 | A wrapper-based method for feature selection to enhance the classification accuracy of Arabic text | Term-frequency inverse document frequency (TF-IDF) for term weighting | SO KNN and the number of selected features | BGWO with elite-based crossover | ✓ (selection) | Decision Tree, Naive Bayes, SVM | 3 Arabic public datasets | Decision tree: F-measure: 0.8255 Naive Bayes: F-measure: 0.93445 SVM: F-measure: 0.9616 |
| [18] 2019 | A wrapper-based method for feature selection to reduce features dimensionality and improving classification accuracy of diabetes diagnosis | Statistical | MO SVM and the number of selected features | NSGA-II, FA, PSO, ICA | ✓ (selection) | SVM | PIMA Indian Type-2 diabetes dataset | Accuracy: 100%, 100%, 98.2% and 94.6% using FA, ICA, NSGA-II and PSO, respectively |

TABLE 1. (Continued.) Summary of surveyed state-of-the-art studies for meta-heuristic based feature selection/fusion methods (SO: single-objective, MO: multi-objective).

| Study | Objectives | Features | Fitness Function | Optimization Technique | Feature Combination | Classifier | Dataset | Performance Measures |
|-----------|--|--|--|--|---------------------|------------|---|---|
| [23] 2019 | Wrapper-based feature selection approach to reduce the number of selected features while maintaining high classification accuracy | Various types of features | SO KNN and the number of selected features | CSSA | ✓ (selection) | KNN | 27 UCI datasets | Accuracy ranges between 60.44% and 97.83% |
| [27] 2018 | Wrapper-based feature selection approach to reduce the number of selected features while maintaining high classification accuracy | Various types of features | SO KNN and the number of selected features | WOA | ✓ (selection) | KNN | 20 UCI datasets | Accuracy ranges between 78.5% and 100% |
| [24] 2018 | A feature selection approach for solving the dimensionality reduction problem of datasets with large-dimensionality and small-instance set | Various types of features | SO KNN and the number of selected features | Hybrid ALO-GWO | ✓ (selection) | KNN | 7 Micro-array gene expression datasets 5 Face detection datasets | Gene Expression: Fitness function ranges between 0 and 0.192 Face Detection: Fitness function ranges between 0 and 0.178 |
| [20] 2018 | A feature selection approach to avoid trapping in local solution when applying to high-dimensional datasets | various types of features | SO KNN and the number of selected features | Asynchronous accelerating multi-leader SSA with 3 different asynchronous updating strategies | ✓ (selection) | KNN | 20 UCI datasets | Accuracy ranges between 73.64% and 99.77% |
| [15] 2018 | A hybrid approach for feature weighting to enhance the diagnostic accuracy of Magnetic Reasoning (MR) brain tumor classification (Cho/Cr)] | Conventional metabolite ratios [Choline/N-acetyl aspartate (Cho/NAA), Cho/Creatine (Cho/Cr)] | SO Fisher criterion | PFree BAT | ✓ (fusion) | KNN | Magnetic resonance spectroscopy (MRS) data of 50 subjects | Sensitivity: 96% Specificity: 91% Accuracy: 93.72% |
| [21] 2018 | Wrapper-based feature selection approach to test the ability of SSA in feature selection problems | Various types of features | SO KNN and the number of selected features | BSSA | ✓ (selection) | KNN | 22 UCI datasets | Accuracy ranges between 73.35% and 100% |
| [22] 2018 | Wrapper-based feature selection approach to reduce the number of selected features while maintaining high classification accuracy | Various types of features | SO KNN and the number of selected features | BSSA with inertia weight parameter | ✓ (selection) | KNN | 23 UCI datasets | Accuracy ranges between 63.19% and 98.99% |
| [30] 2017 | Wrapper-based feature selection approach to enhance the classification accuracy | Various types of features | SO ELM and the number of selected features | SCO | ✓ (selection) | ELM | 10 UCI datasets | Accuracy ranges between 76.80% and 98.73% |
| [17] 2016 | A wrapper-based feature selection approach for improving classification accuracy in liver and kidney disease diagnosis | First order statistical features: Mean, standard deviation, variance, kurtosis, skewness, GLCM: Fourteen second order statistical features | SO misclassification rate of PNN/SVM classifiers | VBBPSO IVbBoPSO | ✓ (selection) | PNN, SVM | Total of 108 Liver abdominal CT image slices and 132 Kidney abdominal CT image slices | Liver using IVbBoPSO: Accuracy: 82.86% Kidney using VbBoPSO: Accuracy: 90% |

ISCA, an accuracy of 76.80%-98.73% was achieved using around 50% of the selected UCI datasets, compared to other optimization algorithms.

Table 1 summarizes the presented exhaustive survey of state-of-the-art studies related to feature selection/fusion approaches based on meta-heuristic algorithms. The feature selection/fusion approaches are classified into the following categories; filter, wrapper, hybrid, single-objective, and multi-objective approaches.

B. CRACK DETECTION TECHNIQUES

Lately, various studies developed promising crack detection systems, however, a limited number considered utilizing feature selection and fusion methods for achieving enhanced recognition.

In [31], Jang *et al.* proposed a hybrid crack detection approach based on deep learning to improve crack detection rate and reduce false alarms. The proposed approach has the power of combining vision and laser IR thermography images. After image reconstruction using Time-spatial-Integrated (TSI) coordinate transform, the proposed approach utilized features extracted from a pre-trained GoogLeNet CNN architecture. The used dataset is a total of 200 raw images increased to 20,000 images via segmentation and augmentation. It was observed that the performance of the proposed crack detection system was improved using hybrid images, specifically, through increasing precision from 59.84% to 98.72% and recall from 97.26% to 99.23%.

In a similar vein, Silva *et al.* [32] developed a concrete crack detection system based on deep learning using transfer learning schema. The proposed system used the pre-trained VGG16 deep learning CNN model. Beside the system development, the authors studied the impact of different training parameters on the performance of the proposed crack detection system. The proposed system was evaluated on a balanced dataset of total 3500 images (intact and crack) of concrete surfaces with 80% used for training and 20% for testing. The obtained experimental results showed that the proposed system achieved an accuracy of 92.27%.

Moreover, Dorafshan *et al.* in [33], proposed a new hybrid crack detector that combined an edge detection method with a DCNN model for enhancing the accuracy of crack detection via reducing the residual noise generated by edge detectors in the final binary images. Moreover, the authors presented a comprehensive comparison between conventional edge detectors and deep learning models trained in three modes for crack detection problem. The dataset used is a total of 100 raw concrete images. Edge detection methods performed well, specifically, the LoG technique that detected about 53-79% of cracked pixels accurately, whereas, it produced noise in the final binary images. For AlexNet, the transfer learning mode achieved an accuracy of about 98% with a slight increase compared to both full training and classification modes, which achieved an accuracy of about 97%. Generally, the hybrid method reduced the noise by a factor of 24.

Also, in [34], Dorafshan *et al.* presented a new benchmark dataset for crack detection called SDNET2018 with benchmark results for crack detection using deep learning model. This dataset includes many challenges such as edges, shadows, scaling and surface roughness. A total of 230 raw images of cracked and intact concrete surfaces of three different types (bridge decks, walls, pavements) were divided into sub-images of size (256 × 256 pixels) resulting into more than 56,000 sub-images used for training, validation, and testing. Then, the AlexNet DCNN architecture was used in both full training (FT) and transfer learning (TL) modes for crack detection. The achieved results showed that using AlexNet in transfer learning mode outperformed the one in full training mode, as the AlexNet FT mode achieved an accuracy of 91.92% against 90.45% for TL mode with bridge deck and an accuracy of 89.31% against 87.54% for TL mode with walls. Also, it achieved an accuracy of 95.52% against 94.86% for TL mode with pavement.

In [35], Maeda *et al.* proposed a CNN based road crack detection approach. The dataset used consists of 500 pavement images captured using a smartphone at the Temple University campus. After segmentation and augmentation, datasets consisting of 640, 000, 160, 000, and 200, 000 samples were used as the training, validation, and testing datasets, respectively. A comparative study was performed between the proposed method, the SVM method, and the Boosting method. The experimental results showed that the proposed ConvNet method outperformed both the SVM and Boosting methods through increasing the F-measure from 0.7359 to 0.8965.

In [36], Cha *et al.* proposed an automatic crack detection method based on DL to minimize the influence of noise caused by different reasons on the classifier. The proposed method used a deep architecture of CNN that is capable of learning features automatically from raw data. Many images of a complex engineering building were captured with several image variations using a DSLR camera. Then, all images were cropped into small batches of size 256 × 256, then divided into training and validation datasets. The results of the experiment proved that the proposed method outperformed the other methods by achieving an accuracy of 97.95%.

In [37], Zhang *et al.* proposed a unified crack detection approach for pavement and sealed cracks using transfer learning. The main advantage of this approach is to solve the difficulty of crack extraction and the inaccurate budgeting resulting from noises and sealed cracks and cracks with similar intensity and width, respectively. A novel two-step pre-classification based on transfer learning was conducted to increase the detection accuracy. After the pre-processing step, a two-step DCNNs model was applied in transfer learning mode to classify images into 3 classes (background, crack, and sealed crack). After that, a thresholding-based segmentation was used to generate the binary image. In order to extract the final crack region, a curve detection method based on tensor voting was applied. Based on the conducted experiments,

the proposed approach achieved a reasonable performance with recall of 0.951 and precision of 0.847.

In [6], Kim *et al.* proposed a ML based crack classification approach to solve the challenge of classifying crack-like patterns as a crack. The proposed approach consisted of two main steps, namely, crack candidate region (CCR) generation and Speeded-Up Robust Features-based (SURF) and CNN based classifications. Image binarization was used to initially extract all crack candidates from images, which accordingly were manually annotated as crack or non-crack. After the annotation, these crack candidates were used in building the classification models. The experimental results showed that the proposed CNN-based crack classification approach achieved an accuracy of 98% and F-measure of 0.95.

Wang *et al.* in [38], proposed an efficient crack detection model combining the strengths of using multiple visual features, such as texture and edges, and the power of multi-task learning. This research aimed to handle the problem of limited representation resulting from using single type of features via combining the LBP and the HOG as two complementary features. The extracted texture and edges feature vectors were combined and fed into a multi-task learning classification based the ELM approach for crack detection. The obtained experimental results showed that the proposed crack detection model achieved an accuracy of 92%, compared to other traditional methods.

In [39], Xu *et al.* presented an end-to-end CNN-based crack detection model to improve the crack detection accuracy via avoiding losing the information of crack edge caused by the process of pooling. The proposed model has the advantage of combining the power of the atrous convolution, Atrous Spatial Pyramid Pooling (ASPP) module, and depth-wise separable convolution in obtaining denser feature map, and multi-scale image feature with avoiding details loss and reducing computational complexity. On evaluating the proposed model using the collected bridge crack images, it achieved a crack detection accuracy, precision, sensitivity, specificity and F-measure of 96.37%, 78.11%, 100%, 95.83%, and 0.8771, respectively. Also, it was observed that the proposed model outperformed the investigated traditional DL models.

Table 2 summarizes the presented exhaustive survey of state-of-the-art studies related to crack detection systems.

III. PRELIMINARIES

A. HISTOGRAM OF ORIENTED GRADIENTS (HOG)

The Histogram of Oriented Gradients (HOG) feature descriptor is one of the most widely used feature descriptor types in computer vision [38], [40], [41]. It can be applied in various pattern recognition domains such as face recognition, pose estimation, human detection etc., with achieving superior results, because of its high ability to strongly describe texture and shape. The HOG descriptor also has the ability of preserving image local information using orientation intensity

distribution and edges gradient. The HOG feature descriptor can be calculated according to Algorithm 1 [38], [40], [41].

Algorithm 1: HOG Feature Descriptor Computation

Input : Grayscale image

Output: Final HOG feature vector $FHOG$

- 1 Calculate Both vertical and horizontal gradients using equations (1) and (2):

$$G_x = I_f * [-1, 0, 1] \quad (1)$$

$$G_y = I_f * [-1, 0, 1]^T \quad (2)$$

- 2 Calculate gradient magnitude and angular orientations using equations (3) and (4):

$$m(x, y) = \sqrt{G_x^2 + G_y^2} \quad (3)$$

$$\theta(x, y) = \tan^{-1} \left(\frac{G_x}{G_y} \right) \quad (4)$$

- 3 Divide the image up into cells and blocks
 - 4 Build the HOG for each cell
 - 5 Compute the normalized HOG for each block
 - 6 Find the final HOG vector by concatenating all blocks histograms
 - 7 **return** $FHOG$
-

B. LOCAL BINARY PATTERNS (LBP)

The Local Binary Patterns (LBP) feature descriptor is widely used as an effective statistical texture descriptor of images in various computer vision systems. The main advantage of the LBP with extracting of image features is robustness to illumination and rotation variation. The LBP feature descriptor can be calculated according to Algorithm 2 [38].

C. CONVOLUTIONAL NEURAL NETWORK (CNN)

Among the various deep neural network models, CNN is considered the most commonly used model for image classification. Standard CNN consists of several convolutional layers, pooling layers, and fully-connected (FC) layers. The main aim of the CNN is the automatic and adaptive learning of spatial hierarchies of useful features, from low-to high-level patterns [32], [36], [42], [43]. Table 3 describes the different CNN layers.

D. BINARY SALP SWARM ALGORITHM (BSSA)

The Binary Salp Swarm Algorithm (BSSA) is one of the most recent meta-heuristic algorithms, mimicking the swarming behavior of salps during the navigation and foraging in deep oceans. It can be used for solving various optimization problems with achieving superior results. In order to simulate the behavior of salp chain mathematically, the population has been divided into two groups, namely, leader salp and follower salps, based on the positions of salps in the population. A salp in the front of the food chain, which is the nearest salp

TABLE 2. Summary of surveyed state-of-the-art studies for crack detection approaches.

| Study | Objectives | Features | Fitness Function | Optimization Technique | Feature Combination | Classifier | Dataset | Performance Measures |
|--------------|--|--|------------------|------------------------|---------------------|--------------|--|--|
| [31] 2019 | A hybrid crack detection approach based on deep learning to improve crack detection rate and reduce false alarms | CNN-learned | × | × | ✓ (Fusion) | Softmax | Total of 200 raw images increased to 20,000 images via segmentation and augmentation | Precision increased from 59.84% to 98.72% Recall increased from 97.26% to 99.23% |
| [6] 2019 | A machine learning-based crack classification approach to solve the challenge of classifying crack-like pattern as a crack | CNN-learned, SURF | × | × | × | Softmax, SVM | Total of 487 raw images collected from concrete surfaces | Accuracy: 98% F-measure: 0.95 |
| [39] 2019 | An end-to-end CNN-based crack detection model to improve the crack detection accuracy via avoiding losing the information of crack edge | CNN-learned features, based atrous convolution, ASPP module and depth-wise separable convolution | × | × | × | Softmax | Total of 2068 bridge crack images collected by the Phantom 4 Pro's CMOS surface array camera | Accuracy: 96.37% Precision: 78.11% Sensitivity: 100% Specificity: 95.83% F-measure: 0.8771 |
| [35] 2018 | A CNN based detection approach for enhancing crack detection accuracy | CNN-learned | × | × | × | Softmax | Total of 500 Pavement images captured using a smart phone at the Temple University campus | F-measure increased from 0.7359 to 0.8965 |
| [38] 2018 | An efficient crack detection model combining the power of using multiple visual features | LBP, HOG | × | × | ✓ (Fusion) | ELM | Total of 44K images generated from raw images of concrete structures (beams, deck slab) captured at Shijiazhuang Tiedao University | Recall rate increased from 89.7% to 89.8% Precision rate increased from 91.9% to 92.3% |
| [32] 2018 | A concrete crack detection system based on deep learning using transfer learning schema Studying the impact of various training parameters on the performance | CNN-learned | × | × | × | Softmax | A balanced dataset of total 3500 images (intact and crack) of concrete surfaces | Accuracy: 92.2% |
| [34] 2018 | A new benchmark dataset for crack detection with benchmark results | CNN-learned | × | × | × | Softmax | SDNET2018 benchmark dataset | FT mode Accuracy: 87.54% TL mode Accuracy: 89.31% |
| [33] 2018 | A new hybrid crack detector combining the edge detector and the DCNN for enhancing accuracy of crack detection | CNN-learned | × | × | × | Softmax | Total of 100 raw images of concrete surfaces | LoG-based cracked pixels detection with noise: Accuracy: 53% to 79% AlexNet-TL: Accuracy: 98% AlexNet-FT, classification modes: Accuracy: 97% |
| [37] 2018 | A unified crack detection approach for pavement and sealed cracks using transfer learning | CNN-learned | × | × | × | Softmax | Total of 800 images (each of $2,000 \times 4,000$ pixels) of crack/sealed crack | Recall: 0.951 Precision: 0.847 |
| [36] 2017 | A DL-based crack detection method to minimize the influence of noise caused by different reasons on the classifier | CNN-learned | × | × | × | Softmax | Total of 332 raw images of a complex engineering building were captured using a DSIR camera | Accuracy: 97.95% |

TABLE 3. Description of CNN architecture.

| Layer | Description |
|---------------|--|
| Convolutional | <ul style="list-style-type: none"> The main component of CNN with an input array of size I, stride step N and a receptive field. It operates by applying three steps, <ul style="list-style-type: none"> (i) An element-by-element multiplications between a sub-array of A and a receptive field (both of size $N \times N$), (ii) Summation of the multiplied values and adding bias to the summed values, (iii) Adding the final values to the output array. The weight values of a receptive field (also often called the filter, or kernel) are initialized randomly [36]. |
| Pooling | <ul style="list-style-type: none"> A down-sampling process and one of the key aspects of CNN Its purpose is to achieve spatial invariance by decreasing the resolution of the input feature map. There are two types of pooling, namely, max and mean pooling [36, 44]. |
| Activation | <ul style="list-style-type: none"> A non-linear transformation function, which widely used in the standard Artificial Neural Network (ANN). There are many different activation functions such as sigmoidal, RELU, etc. It is applied after finishing convolutional operation to enable the CNN to avoid learning trivial linear combinations of the inputs [36, 45]. |
| Softmax | <ul style="list-style-type: none"> A classifier of input images with a layer for prediction located at the end of the CNN model. Any ML algorithm can also be used, however, to date, using softmax function is the most outstanding method [36]. |

to food source, is considered as the leader salp; the rest are the followers. Thereby, the swarm is guided by the leader salp, and the follower salps keep track of each other (and leader salp directly or indirectly) [14], [20]. The leader position is calculated by equation (6):

$$X_j^1 = \begin{cases} F_j + C_1((ub_j - lb_j)C_2 + lb_j), & \text{if } C_3 \geq 0.5, \\ F_j - C_1((ub_j - lb_j)C_2 + lb_j), & \text{if } C_3 < 0.5, \end{cases} \quad (6)$$

where X_j^1 stands for the leader's position and F_j denotes the position vector of food source in the j^{th} dimension, the ub_j and lb_j represent the upper and lower limits of j^{th} dimension, respectively. C_2 and C_3 are parameters with random values inside $[0,1]$, where C_1 is the major parameter of SSA, and is expressed according to equation (7):

$$C_1 = 2e^{-\left(\frac{4t}{Max_{iter}}\right)^2}, \quad (7)$$

where t is the current iteration number and Max_{iter} denotes the maximum iteration number. The location of each follower salp is updated by equation (8):

$$X_j^i = \frac{X_j^i + X_j^{i-1}}{2}, \quad (8)$$

where $i \geq 2$ and X_j^i is the position of the follower salp in the j^{th} dimension.

Since FS is a binary problem, it is supposed for the salps to move in bounded directions (0 and 1 values). In order to convert continuous SSA positions to be suitable for FS problem, a transfer function is used as defined in equation (9), which is the probability of updating an element in the

solution (1 for selected or 0 for not selected).

$$T(X_j^i) = \frac{1}{1 + \exp^{-2(X_j^i)}}, \quad (9)$$

To update an element of a solution in the next iteration, based on the calculated probability from equation (9), equation (10) is used.

$$X_j^i(t+1) = \begin{cases} 1, & \text{if } rand \geq T(X_j^i(t+1)), \\ 0, & \text{if } rand \leq T(X_j^i(t+1)). \end{cases} \quad (10)$$

The pseudo-code of the BSSA algorithm is presented in Algorithm 3 [14], [20].

Due to the high dimensionality of the feature selection problem, BSSA needs some modifications and adaptations to be applied to such a problem. Wherefore, in [20], Aljarah *et al.* proposed an improved BSSA with new operators for dynamically updating the main parameter of BSSA. It also presented a new leadership structure, which assumes half of the population as leaders and the rest as followers instead of a single leader. Then, the whole food chain was divided into several sub-chains with different leaders. In each sub-chain, the salps' positions inside the search space are adaptively updated using a different strategy to strengthen the effectiveness of the BSSA in the matter of exploitative and exploratory tendencies. As a result, according to the best updating strategy called *Termite Colony Salp Swarm Algorithm (TCSSA3)* [20], the major parameter of SSA C_1 is dynamically updated using asynchronous updating rules according to equation (11) instead of decreasing C_1 parameter gradually over the course

Algorithm 2: LBP Feature Descriptor Computation

Input : Grayscale image
Output: Final LBP feature vector *FLBP*

- 1 Divide the image into N cells
- 2 For each pixel in a cell, compare the pixel with its 8 neighbors

$$s(i) = \begin{cases} 1, & \text{if } (g_i - g_c) > 0, \\ 0, & \text{if } (g_i - g_c) \leq 0, \end{cases}$$

where g_c and g_i are the gray level of the cell's center pixel and the surrounding pixels, respectively, resulting in 8-bit binary vector

- 3 Convert each 8-bit binary vector into its corresponding decimal value and replace the intensity value with this decimal value using equation (5):

$$LBP_C = \sum_{i=0}^7 s(i) * 2^i. \quad (5)$$

- 4 Extract the histogram containing only values (0 to 255) for each cell
- 5 Find the final LBP vector by concatenating all histograms of all cells

return *FLBP*

Algorithm 3: Binary Salp Swarm Algorithm (BSSA)

Input : Swarm size n ; Problem dimension d ;
Maximum number of iterations Max_{iter}
Output: The leader salp F

- 1 Initialize the swarm $X_i (i = 1, 2, \dots, n)$
- 2 **while** ($t < Max_{iter}$) **do**
- 3 Obtain the fitness of all salps F =the best search agent
- 4 Set F as the leader salp
- 5 Update C_1 by equation (7)
- 6 **for** (every salp (x_i)) **do**
- 7 **if** ($i=1$) **then**
- 8 Update the position of leader by equation (6)
- 9 Calculate the probabilities using equation (10) that takes the output of equation (9)
- 10 **else**
- 11 Update the position of followers by equation (8)
- 12 **end**
- 13 **end**
- 14 Update the population using the upper and lower limits of variables
- 15 **end**
- 16 **return** The leader salp F

of iterations as in the basic SSA.

$$C_1 = \begin{cases} 1.95 - 2t^{1/3}/Max_{iter}^{1/3}, & \text{if sub-swarm1 or sub-swarm2,} \\ (-2t^3/Max_{iter}^3) + 2.5, & \text{if sub-swarm3 or sub-swarm4.} \end{cases} \quad (11)$$

Then, the salps' positions are updated using equations (6), and (8). The IBSSA (TCSSA3) works, as follows:

- Initialize the population of salps,
- Repeat the following steps until stopping condition is met:
 - Calculate the fitness values of all salps,
 - Set as the best salp, F ,
 - Divide the population of salps into different sub-chains,
 - Update each salp in leaders group, as follows:
 - * Update C_1 by equation (11),
 - * Update the leader's position by equation (6),
 - * Calculate the probability of a feature to be selected using equation (10),
 - Update the follower's position by equation (8),
- Return F .

In this paper, a multi-objective version of the Improved BSSA (MO-IBSSA) with the Kappa index as fitness is proposed. The details of the proposed MO-IBSSA is described in section IV-D2.

E. FILTER-BASED FEATURE SELECTION

Mutual Information (MI) is one of the most widely feature selection methods, used for measuring the mutual dependence between random variables. It therefore provides a way to evaluate the relevance between individual features and classes. The mutual information $I(X; Y)$ between two random variables X and Y can be expressed as in equation (12) [12], [46].

$$I(X; Y) = - \sum_{k,z} p(x_k, y_z) \log_2 p\left(\frac{p(x_k, y_z)}{p(x_k) \cdot p(y_z)}\right), \quad (12)$$

where p is the joint probability mass function of x and y . ReliefF is a ranking approach for features based on a k -nearest neighbor algorithm and can be computed according to equation (13) [12].

$$ReliefF(x_k) = P(x_k \text{ value} | class_d) - P(x_k \text{ value} | class_s), \quad (13)$$

where x_k is the k^{th} feature, $class_d$ and $class_s$ are the different class and the similar class. P is defined as the probability.

Fisher score is a widely used supervised approach for features ranking according to their discriminant ability. Where, it evaluates features comprehensively, considering both minimizing the intra-class distance and maximizing the inter-class distance. Fisher score for the k^{th} feature F_k can be computed according to equation (14) [12], [47].

$$FisherScore(F_k) = \sum_{n=1}^N \left| \frac{\mu_i^k - \mu_j^k}{\sigma_i^{k^2} - \sigma_j^{k^2}} \right|, \quad (14)$$

where μ_i^k and μ_j^k are the mean of the k^{th} feature in the i^{th} and j^{th} classes, and σ_i^k and σ_j^k are the corresponding standard deviation. As it evaluates each feature individually; so, it cannot pay attention to redundancy.

F. SUPPORT VECTOR MACHINE (SVM)

Support Vector Machine is a widely used ML algorithm for classification and regression tasks with superior results. The conventional SVM classifier is defined for binary class problems through maximizing the margin between both classes depending on the training cases placed on the borders. For example, given a training dataset D with n samples $\{x_1, x_2, \dots, x_n\}$, where x_i is a feature vector in a v -dimensional feature space belonging to either of two linearly separable classes C_1 and C_2 . Geometrically, the SVM algorithm finds an optimal decision boundary that achieves the maximal margin separating the samples of the two classes. Achieving this objective requires to solve the optimization problem, defined in eqnarray (15) [48], [49]:

$$\begin{aligned} & \text{maximize} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ & \text{Subject - to : } \sum_{i=1}^n \alpha_i y_i, \quad 0 \leq \alpha_i \leq C, \end{aligned} \quad (15)$$

where, α_i is the assigned weight to the training sample x_i . If $\alpha_i > 0$, x_i is called a support vector. C is a regulation parameter applied to trade-off between the accuracy of training and the complexity of the model to be able to achieve a superior generalization capability. K is a kernel function used to compute similarity between each pair of samples [48], [49].

IV. THE PROPOSED SYSTEM

In this section, we introduce our novel crack detection and crack severity recognition system. The proposed system is composed of two modules. The first one is the feature extraction module based on hand-crafted feature engineering and CNN-based deep feature learning techniques. The second module applies feature selection using hybrid filter-wrapper with MO-IBSSA for reducing feature dimensionality and improving crack severity recognition rate. Figure 1 depicts the general structure of the proposed system for crack detection and severity recognition. The idea behind our proposed system is to improve the crack severity recognition accuracy through using the optimal fused feature set consisted of hand-crafted features or CNN learned features.

A. IMAGE ACQUISITION

In this phase, real data of surface cracks is collected from an ancient building with crack problems. A crack is defined as a damage distinguishable by the human eye. Two primary datasets of 40 and 50 raw images (with and without cracks) of the building surfaces were captured using Canon camera (Canon EOS REBEL T3i) with 5184×3456 resolution over

two years (2018 and 2019) from the historical Mosque (Masjed) of Amir al-Maridani (dating from 1340), located in Cairo - Egypt, with location coordinates 30.03974 N 31.25922 E. That mosque is well-known as one of the famous Islamic architecture historical buildings in Cairo, which is currently under restoration by the European Union.

B. DATA PREPARATION

In this phase, data is prepared and pre-processed for the next training phase. As shown in Figure 1, the data preparation phase consists of multiple steps as follows:

- **step 1: Data-bank generation:** The raw images were cropped into small images of (256×256 pixel resolution) for binary classification and of (128×128 pixel resolution) for multi-class classification, which were then manually annotated as intact or crack images for binary classification and as fine, moderate, and severe crack images for multi-class classification with the help of an expert.
- **step 2: Data cleansing:** The images, including wood patterns, highly illuminated, or have complex shading, which have high potential to trigger false alarms, are neglected.
- **step 3: Image augmentation:** Aiming to enlarge the training dataset via generating new samples similar to the original training samples, data augmentation was applied [31], [50]. In this work, *spatial and Intensity transformations* were used to generate new samples from training samples. Gaussian and salt-and-pepper noise (for only our-dataset-1 and our-dataset-2), flipping, rotation, and a combined transformation were applied for image augmentation in this work, according to the following systematic way:
 - Flipping image vertically.
 - Flipping image horizontally.
 - Flipping image vertically + horizontally.
 - Rotating image by 90.
 - Rotating image by -90.
 - Flipping rotated images horizontally.
 - Adding Gaussian and salt-and-pepper noise.
 - Combining the output images of the previous steps to create the final augmented dataset.

Samples of different cracks and intact types are shown in Figure 2.

C. FEATURE EXTRACTION

In this phase, the proposed system utilizes two types of feature extraction methods for extracting features from the prepared images, including hand-crafted features, using both HOG and LBP methods, and CNN-learned features, using the VGG16 CNN pre-trained model. The output resulted from these methods was used for obtaining three feature vectors, which represent the characteristics of the input images.

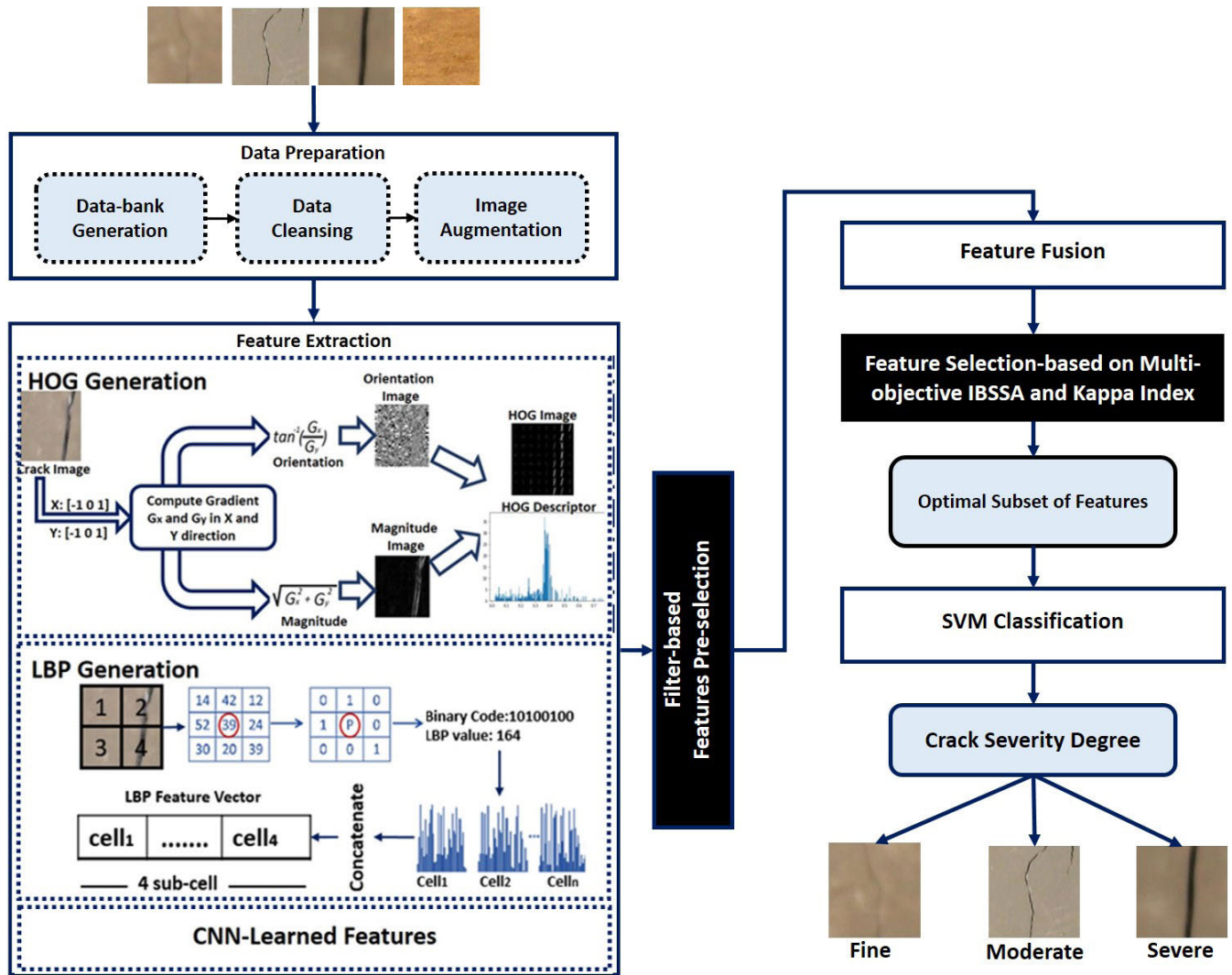


FIGURE 1. Architecture of the proposed crack severity recognition system.

1) CNN FEATURE LEARNING

In this step, the prepared training dataset is fed into a pre-trained VGG16 deep learning model, which is pre-trained with ImageNet dataset as a base network for feature extraction. Extraction and learning of CNN-learned features is achieved in the fully-connected (FC) layer of the VGG16 pre-trained model.

2) HAND-CRAFTED FEATURES ENGINEERING

In this phase, after converting the prepared images from RGB to grayscale, the data of HOG and LBP feature vectors are extracted from the grayscale images according to Algorithm 1 and Algorithm 2, respectively. The hand-crafted features extraction works as illustrated in Algorithm 4.

D. FEATURE FUSION AND SELECTION

This phase consists of two components, namely, feature pre-selection and feature re-selection.

Algorithm 4: Hand-Crafted Feature Vectors Generation

Input : RGB image
Output: HOG feature vector HOG_{hist} ;
 LBP feature vector $LBPhist$

- 1 Convert RGB image into grayscale image $GImg$
- 2 Compute HOG feature vector HOG_{hist} using Algorithm 1
- 3 Divide $GImg$ into 15 non-overlapped regions as shown in Figure 3
- 4 **for** (every region R_i) where, $i = 1, 2, 3, \dots, 15$ **do**
- 5 Compute LBP feature vector $LBPhist_i$ using Algorithm 2
- 6 **end**
- 7 Generate the final $LBPhist$ via concatenating all LBP feature vectors of the 15 regions
- 8 **return** $HOG_{hist}, LBPhist$

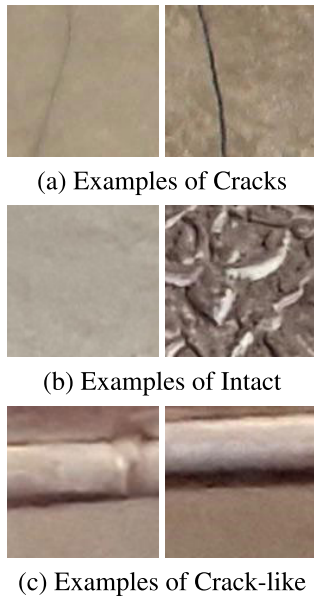


FIGURE 2. Samples of different cracks and intact types.



FIGURE 3. The division process of LBP regions.

1) FEATURE PRE-SELECTION AND FUSION STEP

Using all features, especially the high-dimensional features, as an input to a classifier may consume a lot of computational time while the results may be unsatisfactory. So, in the feature pre-selection and fusion step, a novel filter-based feature pre-selection is applied to pre-select the N^{th} highly ranked features from the original high-dimensional space. These pre-selected features are candidates for the next feature re-selection step. Generally, this feature pre-selection step works according to Algorithm 5.

Algorithm 5: Features Pre-Selection

Input : Training dataset T

Output: Selected features $Feat_{pre}$

- 1 Compute each of mutual information MU , fisher score FS , and ReliefF score RS according to equations (12), (13), and (14) using T
- 2 Select the highest weighted 80% of ranked features MU_r , FS_r , and RS_r according to MU , FS , and RS
- 3 Compute the pre-selected features $Feat_{pre}$ as the intersection among MU_r , FS_r , and RS_r
- 4 **return** $Feat_{pre}$

After selecting the top-ranked features from the strongly relevant features using algorithm 5 for both HOG histogram

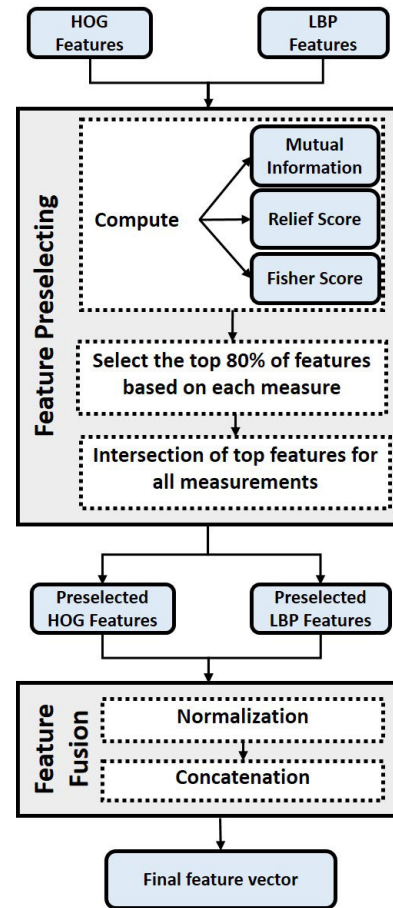


FIGURE 4. Feature pre-selection and fusion stage.

and LBP histogram, a normalization process is performed to transform all feature vectors within a range $[0,1]$, to make the feature vectors compatible with each other when fused later. The feature pre-selection and fusion approach of hand-crafted features is illustrated in Figure 4.

2) FEATURE RE-SELECTION STEP

After conducting the feature pre-selection step, which reduces the original feature dimensionality, another feature re-selection step is carried out for getting the optimal trained SVM (polynomial kernel of degree = 2) model with the highest training and testing Kappa index using a wrapper feature selection method based on MO-IBSSA. The output of this step is the optimal feature subset. As discussed in section III-D, an IBSSA is proposed in [20] to be applied for feature selection problems. In this paper a multi-objective version for IBSSA is developed.

In multi-objective problems, the IBSSA is able to save multiple solutions as the best solution, and update the food source with the best obtained solution so far in every iteration. In order to achieve this goal, the IBSSA is equipped with a repository of food sources for maintaining the best obtained non-dominated solutions so far during optimization and selecting the food source from a group of non-dominated

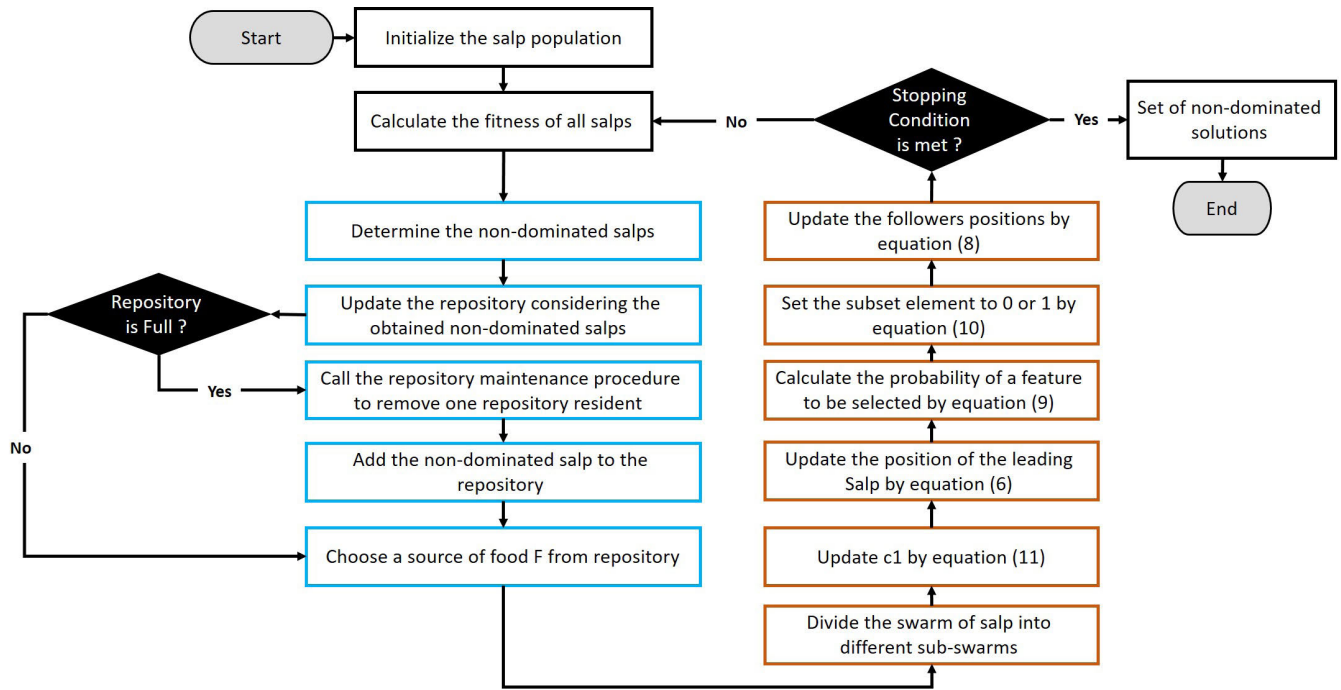


FIGURE 5. Flowchart of the proposed MO-IBSSA.

solutions with the least congested neighborhood using ranking process and Roulette Wheel selection, like the one employed in the repository maintenance operator, but with a different probability of selecting the non-dominated solutions [51]. The workflow of the proposed MO-IBSSA is depicted in Figure 5.

Due to the fact that the Kappa index considers not only the correct rate of a classifier, but also the distribution of per class error, it is utilized in this research as an objective function. It can be defined according to equation (16) [19].

$$\kappa(\ell, D) = \frac{p_o(\ell, D) - p_e(\ell, D)}{1 - p_e(\ell, D)}, \quad (16)$$

where, $p_o(\ell, D)$ and $p_e(\ell, D)$ are the relative observed agreement (similar to accuracy) and the hypothetical probability of chance agreement between the labeled data in the dataset D and the classifier C , respectively. Since the extracted features from crack images are of a very high dimensionality and at sometimes there is a small number of samples, in this research, both training Kappa index and testing Kappa index are used as two objective functions for MO-IBSSA to avoid over-fitting as an alternative to cross-validation approach. That is, applying cross-validation to evaluate all the solutions in the population in all the generations of an optimization algorithm can be quite expensive in computational cost.

The two objective functions are defined as $O_1 = \kappa(\ell, D_{training})$, $O_2 = \kappa(\ell, D_{testing})$. The proposed MO-IBSSA aims to maximize both O_1 and O_2 , since it provides a set of non-dominated solutions. Thus, for comparison and evaluation purposes, a random solution (the solution that achieves

the best balance between the two objective functions) is selected for evaluating performance of the proposed approach with 10 selected UCI datasets, crack detection, and crack severity datasets.

E. CLASSIFICATION

Finally, for classification phase, the proposed approach uses the final optimal subset of features obtained by the employed feature selection method for crack detection and crack severity recognition purposes. Also, it applies SVM algorithm (polynomial kernel with degree = 2) as a classifier.

V. EXPERIMENTAL RESULTS AND DISCUSSION

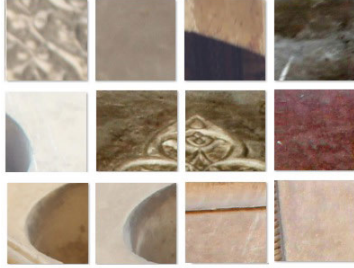
This section presents and discusses all the details related to the experiments carried out to investigate and evaluate the performance of the proposed MO-IBSSA based hybrid filter-wrapper feature selection approach. Moreover, all the details related to evaluating the proposed crack severity recognition system are also described. Simulation experiments were performed on 32 GB RAM, Intel Core i7-4610M CPU (3.00 GHz, 1600 MHz, 4 MB L3 Cache, 2 cores, 37W) and NVIDIA Quadro K5100M 8 GB RAM HP ZBook 15 G2 Mobile Workstation. The proposed approach is designed with Matlab R2015b release on Windows platform.

A. DATASETS AND EVALUATION METRICS

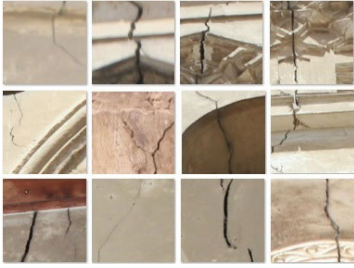
Several experiments were conducted with different crack images datasets as listed in Table 4 using the proposed crack detection and crack severity recognition system. The filter-wrapper feature fusion and selection method based on

TABLE 4. Crack detection datasets.

| NO. | Dataset | No. images | Size of images | No. classes |
|-----|----------------------------|------------|----------------|-------------|
| 1. | our-dataset-1 | 11568 | 256X256 | 2 |
| 2. | our-dataset-2 | 9000 | 256X256 | 2 |
| 3. | our-dataset-3 | 1305 | 128X128 | 3 |
| 4. | bridge-cracks-dataset [39] | 4857 | 128X128 | 2 |



(a) Intact samples



(b) Crack samples

FIGURE 6. Our-dataset-1 samples.

MO-IBSSA is used to reduce feature dimensionality and increase crack detection accuracy.

The first three cracks datasets in Table 4 are expert-based annotated primary datasets of crack images, collected over 2 years from a historical location. Figures 6, 7 and 8 show some samples are selected from the three datasets.

To evaluate the performance of the proposed system, several performance metrics, namely, Accuracy, Recall (Sensitivity), Precision, and F-measure, were calculated according to equations (17)-(21), respectively. In addition, the *Jaccard* similarity score and the *Geometric Mean (GM)* are calculated using equation (22) and equation (23), respectively, where, TP is the true positive values, FP is the false positive values, TN is the true negative values, FN is the false negative values, and N is the total number of observations.

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP}. \quad (17)$$

$$Recall(Sensitivity) = \frac{TP}{TP + FN}. \quad (18)$$

$$Precision = \frac{TP}{TP + FP}. \quad (19)$$

$$Specificity = \frac{TN}{TN + FP}. \quad (20)$$



(a) Intact samples



(b) Crack samples

FIGURE 7. Our-dataset-2 samples.

(a) Fine cracks



(b) Moderate cracks



(c) Severe cracks

FIGURE 8. Our-dataset-3 crack severity samples.

$$F - measure = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}. \quad (21)$$

$$Jaccard = \frac{TP}{TP + FP + FN}. \quad (22)$$

$$GM = \sqrt{Sensitivity * Specificity}. \quad (23)$$

Also, the *Area Under Curve (AUC)* separability measure is calculated. The AUC metric is commonly used as a performance measurement for classification problems using various threshold settings. It represents a measure for model's separability or distinguish-ability between classes.

B. CRACK DETECTION EVALUATION

To evaluate the proposed crack detection and crack severity recognition system, each dataset is randomly split into 70% and 30% of the samples as the training set and testing sets, respectively using hold-out cross-validation. In all

experiments, 5 independent runs are adopted, the population size is set to 20, and the maximum number of generations is set to 100. The filter-based features pre-selection step is first run on the training set to obtain the most relevance, discriminant features and to reduce the dimensionality for the wrapper-based feature selection approach. Then, the performance of the optimal feature subsets is evaluated by the learning/classification algorithm on the testing set. The SVM classifier, with polynomial kernel (degree = 2) is selected as the learning algorithm due to its superior results in the field of image classification.

For crack detection datasets (1, 2, 4), the average and standard deviation of the fitness and reduction rate, obtained using the subsets of features selected by the proposed system as the training and testing datasets over the 5 independent runs, are presented in Table 5. In addition, the average running time is also shown. For crack severity recognition dataset (3), two types of features, namely, the fused hand-crafted features and CNN-learned features extracted from VGG16 deep learning model are used for recognition purposes. The average and standard deviation of the attained fitness and reduction rate for the training and testing datasets over the 5 independent runs are also presented.

Tables 6 presents the accuracy and F-measure of classification using the whole raw features as well as the accuracy and F-measure of classification using the selected subset of features. From Table 6, it is noticed that the proposed crack detection and severity recognition system outperformed the traditional classification, which doesn't apply feature selection, for the 3 crack detection datasets (1, 2, 4). Whereas, it increased the detection rate by approximately 27%, 27%, and 57% for bridge-cracks-dataset [39], our-dataset-2, and our-dataset-1, respectively. It also reduced the features by approximately 64.68%-68.8% for all cases.

For crack severity dataset (3), the proposed system improved the recognition rate by 5.84% using the fused hand-crafted features. While, there is a slight degradation in performance by only 0.44% using the learned features, but with feature reduction rate of approximately 54.9%.

Tables 7 and 8 present the different performance metrics mentioned above, which are used to evaluate the performance of the proposed crack detection and severity recognition.

In Table 7, the performance of the proposed system is illustrated considering our-dataset-1, our-dataset-2, and bridge-cracks-dataset [39]. For our-dataset-1, it was noticed that the achieved accuracies were 96.27% and 89.82% for internal and external testing datasets, respectively. Moreover, the obtained recall percentages were 84.97% and 76.36% as a result of 509 TP hits out of 599 actual crack images and 884.2 TP hits out of 1158 actual crack images for internal and external testing datasets, respectively as shown in Figure 9 (I).

Also, for our-dataset-2, it was noticed that the attained accuracies were 96.86% and 93.85% for internal and external testing datasets, respectively. Moreover, the achieved recall percentages were 93.52% and 92.84% as a result of 1083 TP hits out of 1158 actual crack images and 591.4 TP hits out

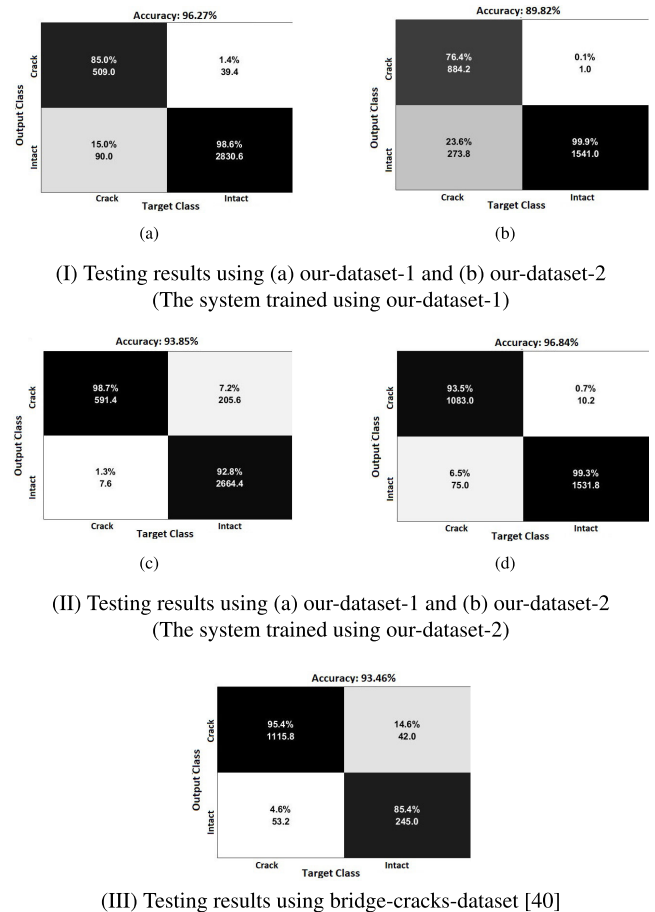


FIGURE 9. Average confusion matrix of the correct classification percentage for all crack detection datasets using the proposed system.

of 599 actual crack images for internal and external testing datasets, respectively, as shown in Figure 9 (II).

For bridge-cracks-dataset [39], it was observed that the obtained accuracy was 93.46%. Moreover, the achieved recall percentage was 85.37% as a result of only 1115.8 TP hits out of 1169 actual crack images, as shown in Figure 9 (III). Accordingly, it was concluded from Table 6 that the proposed system in general improved the accuracy by $\approx 27\%$ - 57% and the F-measure by $\approx 23\%$ - 42% compared to the traditional classification approaches that use the whole set of features with an accuracy of 93.46% to 96.84% with both Crack and Intact classes and with a discriminatory power AUC of 94.19% to 98.42%. Table 8 shows the performance of the proposed crack detection and severity recognition system considering our-dataset-3 using both fused hand-crafted and VGG16 learned features.

For crack severity recognition dataset (3), it was noticed from Table 8 that accuracies of 61.85% and 80.41% were obtained using fused hand-crafted and VGG16 learned features, respectively. Moreover, the achieved recall percentages were 62.15% and 80.52% using fused hand-crafted and VGG16 learned features, respectively.

TABLE 5. Results of implementing the proposed hybrid filter-wrapper feature selection based on MO-IBSSA on crack detection datasets.

| | | Our-dataset-1 | Our-dataset-2 | bridge-cracks-dataset [39] dataset | Our-dataset-3 LBP-HOG | Our-dataset-3 VGG16 |
|-------------------|-----|---------------|---------------|---------------------------------------|--------------------------|------------------------|
| Time (sec) | | 44,108 | 59,252 | 36,560 | 47,708 | 64,720 |
| Training Kappa | AVG | 0.9996 | 1 | 1 | 0.9984 | 0.9997 |
| | STD | 0.0006 | 0 | 0 | 0 | 0.0001 |
| Testing Kappa | AVG | 0.8649 | 0.9351 | 0.7985 | 0.4088 | 0.7012 |
| | STD | 0.0055 | 0.0008 | 0.0029 | 0.0517 | 0.0032 |
| Reduction Rate | AVG | 68.627 | 68.038 | 64.6823 | 68.029 | 54.912 |
| | STD | 0.97807 | 1.3832 | 0.5188 | 0.9584 | 1.9753 |

TABLE 6. Comparison between classification using all raw features and classification using the selected features.

| Without Feature Selection | | | | | | With Feature Selection | | | | |
|---------------------------|----------------------------|-----------------------------------|-----------------------------------|------------------------|--------------------------|----------------------------|-----------------------------------|-----------------------------------|------------------------|--------------------------|
| | bridge-cracks-dataset [40] | our-dataset-1 Internal testing | our-dataset-2 Internal testing | our-dataset-3 VGG16 | our-dataset-3 LBP-HOG | bridge-cracks-dataset [39] | our-dataset-1 Internal testing | our-dataset-2 internal testing | our-dataset-3 VGG16 | our-dataset-3 LBP-HOG |
| Accuracy | 0.6641 | 0.3857 | 0.6974 | 0.8077 | 0.5590 | 0.9346 | 0.9627 | 0.9684 | 0.8041 | 0.6185 |
| F-measure | 0.5356 | 0.4112 | 0.7158 | 0.8069 | 0.5558 | 0.8373 | 0.8320 | 0.9477 | 0.8050 | 0.6138 |
| No. features | 4485 | 6069 | 6069 | 4096 | 7941 | 1584 | 1904 | 1939.8 | 1846.8 | 2538.8 |

TABLE 7. Statistical measures (average and standard deviation) of the performance metrics for the proposed hybrid filter-wrapper feature selection approach based on MO-IBSSA algorithm on crack detection testing datasets over 5 independent runs.

| | | bridge-cracks-dataset [39] | | our-dataset-1 | | our-dataset-2 | |
|----------------|-----|----------------------------|--------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| | | | | Internal testing dataset | External testing dataset | Internal testing dataset | External testing dataset |
| Accuracy | AVG | | 0.9346 | 0.9627 | 0.8982 | 0.9684 | 0.9385 |
| | STD | | 0.0028 | 0.0016 | 0.0018 | 0.0001 | 0.0012 |
| F-measure | AVG | | 0.8373 | 0.8872 | 0.8655 | 0.9622 | 0.9615 |
| | STD | | 0.0072 | 0.0048 | 0.0026 | 0.0001 | 0.0001 |
| Precision | AVG | | 0.8216 | 0.9282 | 0.9989 | 0.9907 | 0.9972 |
| | STD | | 0.0075 | 0.0066 | 0 | 0 | 0.0001 |
| Recall | AVG | | 0.8537 | 0.8497 | 0.7636 | 0.9352 | 0.9284 |
| | STD | | 0.0102 | 0.0033 | 0.0041 | 0.0011 | 0.0015 |
| Specificity | AVG | | 0.9545 | 0.9863 | 0.9994 | 0.9934 | 0.9873 |
| | STD | | 0.0022 | 0.0033 | 0 | 0 | 0.0001 |
| Geometric Mean | AVG | | 0.9027 | 0.9155 | 0.8735 | 0.9639 | 0.9574 |
| | STD | | 0.0056 | 0.0023 | 0.0024 | 0.0001 | 0.0001 |
| Jaccard | AVG | | 0.7202 | 0.7973 | 0.7629 | 0.9271 | 0.9259 |
| | STD | | 0.0106 | 0.0018 | 0.0041 | 0.0001 | 0.0014 |
| AUC | AVG | | 0.9419 | 0.9581 | 0.9486 | 0.9842 | 0.9752 |
| | STD | | 0.0144 | 0.0139 | 0.0187 | 0.0016 | 0.0018 |

It was noticed from Figure 10 (a) that the proposed severity recognition system can recognize crack severity by an accuracy of 67.8%, 54.1%, and 61.7% for fine, moderate, and severe crack, respectively. It was also noticed that a high confusion was shown between moderate and severe classes with an average of approximately 31.5%, between fine and moderate classes with an average of approximately 16.3%, and finally, between fine and severe classes with an average of approximately 10.4% and an AUC of 0.9287. On the other hand, as shown in Figure 10 (b), using VGG16 learned features improved the performance by 17.7%, 15.9%, and 23.5% for fine, moderate, and severe crack, respectively. While, the confusion between crack severity recognition is minimized by 15.2%, 4.3%, and 9.3% between (moderate

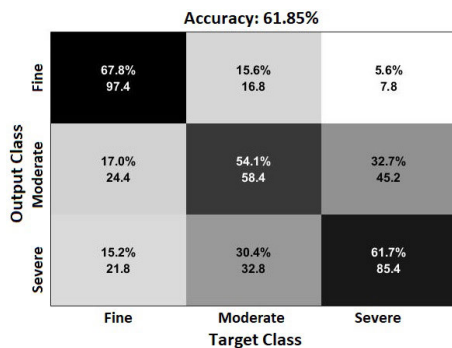
and severe cracks), (moderate and fine cracks), and (fine and severe cracks), respectively.

C. FEATURE SELECTION EVALUATION

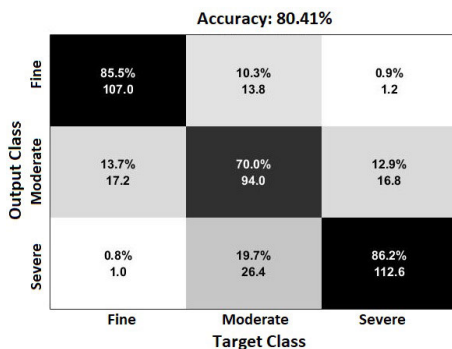
For more validation and to investigate the performance of the proposed feature selection approach, we tested 10 representative datasets obtained from the UCI Machine Learning Datasets Repository [52], as listed in Table 9, including different numbers of features, samples, and classes. The baseline approaches based on the single objective function described in equation (24), which are performed on the same datasets, is briefly presented, as the results obtained by the proposed hybrid filter-wrapper feature selection method will be

TABLE 8. Statistical measures (average and standard deviation) of the performance metrics for the proposed hybrid filter-wrapper feature selection approach based on MO-IBSSA algorithm on crack severity testing datasets over 5 independent runs.

| | | our-dataset-3 VGG16 | our-dataset-3 LBP-HOG |
|----------------|-----|------------------------|--------------------------|
| Accuracy | AVG | 0.8041 | 0.6185 |
| | STD | 0.0067 | 0.0102 |
| F-measure | AVG | 0.8050 | 0.6138 |
| | STD | 0.0070 | 0.0102 |
| Precision | AVG | 0.8059 | 0.6131 |
| | STD | 0.0075 | 0.0101 |
| Recall | AVG | 0.8052 | 0.6215 |
| | STD | 0.0067 | 0.0107 |
| Specificity | AVG | 0.9022 | 0.8088 |
| | STD | 0.0033 | 0.0054 |
| Geometric Mean | AVG | 0.8030 | 0.6049 |
| | STD | 0.0072 | 0.0088 |
| Jaccard | AVG | 0.6783 | 0.4502 |
| | STD | 0.0094 | 0.0095 |
| AUC | AVG | 0.9287 | 0.7638 |
| | STD | 0.0049 | 0.0130 |



(a) Crack severity recognition using LBP-HOG fused features



(b) Crack severity recognition using VGG16 learned features

FIGURE 10. Average confusion matrix of the correct classification percentage of crack severity recognition datasets using the proposed system.

compared with those obtained by these baseline approaches.

$$Fitness = \alpha.E + \beta\left(\frac{R}{N}\right) \quad (24)$$

where E is the classification error rate, R represents the number of selected feature subset and N represents the total

TABLE 9. Selected UCI datasets.

| NO. | Dataset | No. features | No. instances | No. classes |
|-----|--------------|--------------|---------------|-------------|
| 1. | HeartEW | 13 | 270 | 2 |
| 2. | SonarEW | 60 | 208 | 2 |
| 3. | Australian | 14 | 690 | 2 |
| 4. | IonosphereEW | 34 | 351 | 2 |
| 5. | WineEW | 13 | 178 | 3 |
| 6. | CleanI | 166 | 476 | 2 |
| 7. | Leukemia | 7128 | 72 | 2 |
| 8. | Colon | 2000 | 62 | 2 |
| 9. | Movements | 90 | 360 | 14 |
| 10. | German | 24 | 1000 | 2 |

TABLE 10. Parameters setting of the utilized meta-heuristic optimization algorithms.

| Parameter | Value |
|--|--|
| Population size | 20 |
| Max no. of iterations | 100 |
| No. of runs | 20 |
| Problem dimensions | No. of features in the data |
| Search domain | [0 1] |
| Inertia factor of PSO | 0.1 |
| r_1 and r_2 in GWO | both drawn from the uniform distribution |
| α parameter in the fitness function | 0.99 |
| β parameter in the fitness function | 0.01 |

number of features in the dataset, α and β are two constants controlling the importance of classification performance and the length of features subset.

All the tabulated evaluations of the proposed method are recorded and compared to state-of-the-art methods using an HP ZBook 15 G2 Mobile Workstation with 32 GB RAM, Intel Core i7-4610M CPU (3.00 GHz, 1600 MHz, 4 MB L3 Cache, 2 cores, 37W), and NVIDIA Quadro K5100M 8 GB RAM. The proposed approach is designed and developed using Matlab R2015b release on Windows 10 64-bit platform. To have fair comparisons, all methods are developed and tested using MATLAB and by the same computing platform in order to use the same global settings for all algorithms. That is, all used algorithms are uniformly randomly initialized. Besides, all algorithms have a population size of 20 search agents and a number of iterations = 100. These values are selected after conducting several initial empirical studies. Moreover, the average and standard deviation of fitness, and reduction rate, in addition to the average computational time over 20 independent runs are used for comparison purposes.

Generally, to provide a credible evidence that the system has a good generalization performance, k-fold cross-validation was used to assess the processes of feature selection and training [12]. Thus, all baseline approaches were assessed using 5-fold cross-validation to be compared against the proposed feature selection approach and to highlight the effectiveness of the proposed approach without using k-fold cross-validation. On the other hand, to evaluate the proposed feature selection approach, each dataset is randomly split into 70% of the samples as the training set and 30% of the samples as testing set using hold-out cross-validation.

TABLE 11. Statistical measures (average and standard deviation) of the obtained fitness and reduction rate for all base-line algorithms over 20 independent runs.

| | | Dataset | HeartEW | SonarEW | Ionosphere | WineW | Clean1 | Colon | Movement | German | Austurlian | Leukemia |
|---------------|------------|---------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| BGWO | Time (sec) | AVG | 114.31 | 113.14 | 72.84 | 123.52 | 119.88 | 118.02 | 96.699 | 90.69 | 121.93 | 275.16 |
| | | STD | | | | | | | | | | |
| | Fitness | AVG | 0.1809 | 0.1084 | 0.0842 | 0.0357 | 0.0951 | 0.1549 | 0.1625 | 0.2590 | 0.1557 | 0.0187 |
| | | STD | 0.0168 | 0.0151 | 0.0133 | 0.0134 | 0.0091 | 0.0245 | 0.0085 | 0.0132 | 0.0178 | 0.0096 |
| | Red. Rate | AVG | 50.77 | 40.17 | 47.65 | 49.23 | 30.51 | 36.34 | 37.67 | 38.33 | 63.21 | 38.74 |
| | | STD | 6.79 | 6.79 | 10.08 | 9.13 | 4.74 | 4.03 | 6.33 | 9.23 | 14.89 | 2.37 |
| BPSO | Time (sec) | AVG | 104.13 | 100.32 | 113.04 | 114.70 | 81.23 | 89.1780 | 128.24 | 112.05 | 112.07 | 182.04 |
| | | STD | | | | | | | | | | |
| | Fitness | AVG | 0.1836 | 0.1041 | 0.0753 | 0.0340 | 0.0896 | 0.1376 | 0.1569 | 0.2553 | 0.1399 | 0.0126 |
| | | STD | 0.00434 | 0.0115 | 0.00445 | 0.0043 | 0.00677 | 0.0177 | 0.0047 | 0.00425 | 0.0019 | 0.0097 |
| | Red. Rate | AVG | 55.38 | 56.08 | 68.68 | 58.08 | 49.67 | 51.34 | 54.89 | 49.17 | 81.07 | 51.17 |
| | | STD | 5.35 | 4.99 | 5.07 | 8.45 | 3.56 | 1.34 | 5.66 | 6.71 | 8.7558 | 0.6259 |
| BSSA | Time (sec) | AVG | 92.32 | 106.21 | 100.97 | 106.88 | 142.81 | 131.48 | 105.56 | 121.08 | 132.84 | 229.07 |
| | | STD | | | | | | | | | | |
| | Fitness | AVG | 0.2743 | 0.1533 | 0.1198 | 0.0814 | 0.1106 | 0.1854 | 0.1772 | 0.2851 | 0.2214 | 0.0297 |
| | | STD | 0.0186 | 0.0095 | 0.0063 | 0.0092 | 0.0032 | 0.0082 | 0.0038 | 0.0047 | 0.0240 | 0.0048 |
| | Red. Rate | AVG | 31.54 | 24.42 | 38.09 | 40.38 | 12.11 | 12.26 | 20.56 | 27.71 | 54.64 | 16.08 |
| | | STD | 8.24 | 16.84 | 18.19 | 13.64 | 2.55 | 0.76 | 15.87 | 18.05 | 18.87 | 11.94 |
| Improved BSSA | Time (sec) | AVG | 90.49 | 67.05 | 78.95 | 80.94 | 142.24 | 192.43 | 68.99 | 116.13 | 77.83 | 540.93 |
| | | STD | | | | | | | | | | |
| | Fitness | AVG | 0.2106 | 0.1351 | 0.1036 | 0.0438 | 0.1062 | 0.1786 | 0.1722 | 0.2657 | 0.1777 | 0.0252 |
| | | STD | 0.0111 | 0.0071 | 0.0080 | 0.0071 | 0.0063 | 0.0169 | 0.0052 | 0.0064 | 0.0197 | 0.0075 |
| | Red. Rate | AVG | 52.31 | 33.83 | 45.59 | 47.31 | 22.50 | 35.43 | 33.94 | 37.08 | 59.64 | 32.27 |
| | | STD | 5.35 | 7.36 | 9.75 | 7.18 | 13.42 | 9.93 | 7.42 | 8.65 | 16.75 | 13.18 |

TABLE 12. Statistical measures (average and standard deviation) of the obtained fitness and reduction rate for the proposed hybrid filter-wrapper feature selection approach based on MO-IBSSA algorithm over 20 independent runs.

| | | HeartEW | SonarEW | Ionosphere | WineW | Clean1 | Colon | Movement | German | Austurlian | Leukemia |
|----------------|-----|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------|
| Time (sec) | | 43.02 | 35.61 | 38.63 | 37.76 | 91.45 | 53.81 | 61.44 | 100.62 | 48.07 | 99.80 |
| Training Kappa | AVG | 0.6978 | 0.7776 | 0.8277 | 0.9238 | 0.8750 | 0.8063 | 0.8626 | 0.5263 | 0.7596 | 0.9587 |
| | STD | 0.0387 | 0.0466 | 0.0284 | 0.0218 | 0.0153 | 0.0192 | 0.0086 | 0.0295 | 0.0199 | 0.0097 |
| Testing Kappa | AVG | 0.7256 | 0.6369 | 0.7823 | 0.9680 | 0.6959 | 0.9700 | 0.8128 | 0.2375 | 0.6906 | 1 |
| | STD | 0.0074 | 0.0863 | 0.0709 | 0.0209 | 0.0484 | 0.0533 | 0.0101 | 0.0594 | 0.0507 | 0 |
| Reduction Rate | AVG | 68.8462 | 56.1667 | 77.5000 | 63.4615 | 62.1988 | 65.9400 | 63.1667 | 67.2917 | 67.8571 | 60.9827 |
| | STD | 8.0774 | 7.7441 | 5.5045 | 8.2302 | 3.5010 | 3.5688 | 3.4629 | 6.5219 | 5.9084 | 10.4894 |

For comparison purposes, we used four different algorithms, namely, BPSO, BGWO, BSSA, and IBSSA. The details of parameters of these algorithms are presented in Table 10. The values of listed parameters in Table 10 have been selected based on both initial experiments and previous researches [21], [26], [29].

The filter approaches were first run on the training set as a pre-selection step to obtain the most relevance, discriminant features and to reduce the problem dimensionality for the wrapper feature selection approach. Then, the performance of the optimal feature subsets was evaluated by the learning/classification algorithm on the testing set. Due to its simplicity and popularity, the learning algorithm is selected as the KNN, where K is set to 5 in the experiments.

The average and standard deviation of the obtained fitness and reduction rate for all the base-line algorithms and the proposed MO-IBSSA over the 20 independent runs are illustrated in Tables 11 and 12. In addition, the average running time is also presented.

According to Tables 11 and 12, it can be observed that the proposed hybrid filter-wrapper feature selection method based on MO-IBSSA achieved higher performance and feature reduction rate over 70% and 90% of the used datasets, respectively. Also, based on the presented results, it can be noticed that the proposed feature selection method can generally evolve a small number of features and achieve similar or better classification performance compared to all the other optimization algorithms except for Austurlian, Colon cancer and SonarEW datasets. For Austurlian dataset, BPSO has a slight improvement in terms of accuracy by only 0.79% at the expense of increasing the average running time by more than twice. While BGWO has an improvement in terms of accuracy by 2.42% with longer running time by approximately 3 times and more features by 16% for SonarEW dataset. For Colon cancer dataset, the basic Salp optimization algorithm has a very slight improvement by only 0.28% at the expense of increasing the number of features by 53.68% and the average running time by more than twice.

TABLE 13. Performance metrics of the proposed hybrid filter-wrapper feature selection approach based on MO-IBSSA on UCI testing datasets over the 20 independent runs.

| | | HeartEW | SonarEW | Ionosphere | WineEw | Clean1 | Colon | Movement | German | Austurlian | Leukemia |
|----------------|-----|---------------|---------|---------------|---------------|---------------|--------|---------------|---------------|------------|----------|
| Accuracy | AVG | 0.8654 | 0.8210 | 0.9053 | 0.9788 | 0.8475 | 0.9861 | 0.8814 | 0.7073 | 0.8476 | 1 |
| | STD | 0.0038 | 0.0418 | 0.0291 | 0.0138 | 0.0247 | 0.0247 | 0.0189 | 0.0220 | 0.0248 | 0 |
| F-measure | AVG | 0.8821 | 0.7913 | 0.8495 | 0.9798 | 0.8551 | 0.9891 | 0.8784 | 0.8035 | 0.8642 | 1 |
| | STD | 0.0042 | 0.0570 | 0.0531 | 0.0136 | 0.0225 | 0.0193 | 0.0203 | 0.0155 | 0.0215 | 0 |
| Precision | AVG | 0.8591 | 0.8637 | 0.9660 | 0.9783 | 0.9185 | 1 | 0.9010 | 0.7580 | 0.8532 | 1 |
| | STD | 0.0085 | 0.0523 | 0.0421 | 0.0148 | 0.0205 | 0 | 0.0219 | 0.0150 | 0.0263 | 0 |
| Recall | AVG | 0.9067 | 0.7362 | 0.7622 | 0.9825 | 0.8006 | 0.9792 | 0.8814 | 0.8552 | 0.8759 | 1 |
| | STD | 0.0155 | 0.0911 | 0.0772 | 0.0114 | 0.0373 | 0.0370 | 0.0189 | 0.0240 | 0.0255 | 0 |
| Specificity | AVG | 0.8139 | 0.8955 | 0.9843 | 0.9898 | 0.9081 | 1 | 0.9915 | 0.3622 | 0.8125 | 1 |
| | STD | 0.0159 | 0.0434 | 0.0208 | 0.0066 | 0.0246 | 0 | 0.0014 | 0.0512 | 0.0375 | 0 |
| Geometric Mean | AVG | 0.8589 | 0.8099 | 0.8649 | 0.9860 | 0.8524 | 0.9894 | 0.9303 | 0.5550 | 0.8434 | 1 |
| | STD | 0.0036 | 0.0481 | 0.0450 | 0.0091 | 0.0243 | 0.0189 | 0.0112 | 0.0422 | 0.0258 | 0 |
| Jaccard | AVG | 0.7891 | 0.6581 | 0.7418 | 0.9609 | 0.7477 | 0.9792 | 0.8007 | 0.6718 | 0.7614 | 1 |
| | STD | 0.0068 | 0.0766 | 0.0771 | 0.0262 | 0.0390 | 0.0370 | 0.0271 | 0.0214 | 0.0329 | 0 |
| AUC | AVG | 0.9015 | 0.8731 | 0.9460 | 0.9493 | 0.9205 | 0.9903 | 0.9031 | 0.6754 | 0.8885 | 1 |
| | STD | 0.0023 | 0.0332 | 0.0215 | 0.0196 | 0.0138 | 0.0199 | 0.0062 | 0.0283 | 0.0176 | 0 |

TABLE 14. Performance metrics of the BPSO algorithm with the UCI testing datasets over 20 independent runs.

| | | HeartEW | SonarEW | Ionosphere | WineEw | Clean1 | Colon | Movement | German | Austurlian | Leukemia |
|----------------|-----|---------|---------|------------|--------|--------|--------|----------|--------|---------------|----------|
| Accuracy | AVG | 0.8648 | 0.7661 | 0.8648 | 0.9740 | 0.8257 | 0.9389 | 0.8781 | 0.6947 | 0.8553 | 0.9381 |
| | STD | 0.0194 | 0.0264 | 0.0194 | 0.0168 | 0.0241 | 0.0647 | 0.0171 | 0.0171 | 0.0080 | 0.0349 |
| F-measure | AVG | 0.8820 | 0.7177 | 0.8820 | 0.9757 | 0.8277 | 0.9534 | 0.8757 | 0.7957 | 0.8644 | 0.9532 |
| | STD | 0.0183 | 0.0398 | 0.0183 | 0.0167 | 0.0263 | 0.0494 | 0.0167 | 0.0133 | 0.0044 | 0.0260 |
| Precision | AVG | 0.8551 | 0.8243 | 0.9829 | 0.9745 | 0.9323 | 0.9628 | 0.8911 | 0.7480 | 0.8998 | 0.9645 |
| | STD | 0.0133 | 0.0483 | 0.0196 | 0.0175 | 0.0233 | 0.0502 | 0.0184 | 0.0084 | 0.0247 | 0.0364 |
| Recall | AVG | 0.9111 | 0.6397 | 0.6527 | 0.9780 | 0.7450 | 0.9458 | 0.8781 | 0.8502 | 0.8325 | 0.9429 |
| | STD | 0.0306 | 0.0618 | 0.0698 | 0.0152 | 0.0379 | 0.0621 | 0.0171 | 0.0238 | 0.0145 | 0.0293 |
| Specificity | AVG | 0.8069 | 0.8773 | 0.9933 | 0.9871 | 0.9298 | 0.9250 | 0.9913 | 0.3317 | 0.8837 | 0.9286 |
| | STD | 0.0191 | 0.0411 | 0.0076 | 0.0084 | 0.0252 | 0.1008 | 0.0012 | 0.0268 | 0.035 | 0.0733 |
| Geometric Mean | AVG | 0.8573 | 0.7477 | 0.8040 | 0.9824 | 0.8320 | 0.9343 | 0.9299 | 0.5305 | 0.8574 | 0.9350 |
| | STD | 0.0181 | 0.0322 | 0.0416 | 0.0119 | 0.0239 | 0.0718 | 0.0097 | 0.0212 | 0.0104 | 0.0429 |
| Jaccard | AVG | 0.7894 | 0.5611 | 0.6446 | 0.9533 | 0.7068 | 0.9149 | 0.7908 | 0.6609 | 0.7612 | 0.9117 |
| | STD | 0.0286 | 0.0476 | 0.0653 | 0.0307 | 0.0379 | 0.0874 | 0.0256 | 0.0183 | 0.0068 | 0.0482 |
| AUC | AVG | 0.9077 | 0.8440 | 0.9278 | 0.9393 | 0.9300 | 0.9656 | 0.9347 | 0.6678 | 0.8825 | 0.9929 |
| | STD | 0.0096 | 0.0200 | 0.0370 | 0.0015 | 0.0134 | 0.0419 | 0.0092 | 0.0143 | 0.00024 | 0.0045 |

In terms of reduction rate, the proposed hybrid filter-wrapper feature selection method based on MO-IBSSA is in the first place, where it has the lowest number of features over 90% of the used datasets. BPSO comes in second place, followed by BGWO then Improved Salp optimization algorithm, and finally, the basic Salp optimization algorithm in the last place. In terms of average time, the proposed method has the shortest average running time over 90% of the used datasets. Only for Clean-1 dataset, BPSO beats the proposed method. The proposed method minimizes the running time by more than 50% for the most of used datasets. As, comparing the proposed method with other baseline approaches, it is observed that the proposed method can reduce the computational time at least a half compared with other approaches in most cases.

Tables 13-17, present the different performance metrics mentioned above, which are used to evaluate the different algorithms on the testing dataset.

The average and standard deviation of the different performance metrics for the achieved results from performing the proposed hybrid filter-wrapper feature selection method based on MO-IBSSA on 10 UCI datasets over the 20 independent runs are illustrated in Table 13.

The average and standard deviation of the different performance metrics for the attained results from performing the BPSO feature selection method on 10 UCI datasets over the 20 independent runs are illustrated in Table 14.

The average and standard deviation of the different performance metrics for the obtained results from performing the BGWO feature selection method on 10 UCI datasets over the 20 independent runs are illustrated in Table 15.

TABLE 15. Performance metrics of the BGWO algorithm with the UCI testing datasets over 20 independent runs.

| | | HeartEW | SonarEW | Ionosphere | WineW | Clean1 | Colon | Movement | German | Austurlian | Leukemia |
|----------------|-----|---------|---------------|------------|--------|--------|--------|----------|--------|------------|----------|
| Accuracy | AVG | 0.8346 | 0.8452 | 0.8875 | 0.9462 | 0.8415 | 0.8083 | 0.8714 | 0.6620 | 0.8194 | 0.9524 |
| | STD | 0.0217 | 0.0400 | 0.0248 | 0.0118 | 0.0167 | 0.0555 | 0.0133 | 0.0227 | 0.0339 | 0 |
| F-measure | AVG | 0.8546 | 0.8253 | 0.8215 | 0.9486 | 0.8512 | 0.8661 | 0.8697 | 0.7624 | 0.8431 | 0.9630 |
| | STD | 0.0214 | 0.0494 | 0.0465 | 0.0115 | 0.0178 | 0.0427 | 0.0135 | 0.0175 | 0.0274 | 0 |
| Precision | AVG | 0.8338 | 0.8685 | 0.9360 | 0.9465 | 0.9025 | 0.8065 | 0.8860 | 0.7503 | 0.8174 | 1 |
| | STD | 0.0129 | 0.0430 | 0.0365 | 0.0119 | 0.0146 | 0.0320 | 0.0138 | 0.0159 | 0.0546 | 0 |
| Recall | AVG | 0.8767 | 0.7897 | 0.7351 | 0.9518 | 0.8063 | 0.9375 | 0.8714 | 0.7755 | 0.8754 | 0.9286 |
| | STD | 0.0318 | 0.0766 | 0.0688 | 0.0112 | 0.0336 | 0.0709 | 0.0133 | 0.0269 | 0.0533 | 0 |
| Specificity | AVG | 0.7819 | 0.8939 | 0.9716 | 0.9727 | 0.8871 | 0.5500 | 0.9908 | 0.3972 | 0.7500 | 1 |
| | STD | 0.0136 | 0.0374 | 0.0167 | 0.0062 | 0.0203 | 0.0784 | 0.0009 | 0.0468 | 0.0937 | 0 |
| Geometric Mean | AVG | 0.8279 | 0.8390 | 0.8441 | 0.9620 | 0.8454 | 0.7161 | 0.9262 | 0.5540 | 0.8076 | 0.9636 |
| | STD | 0.0202 | 0.0430 | 0.0397 | 0.0087 | 0.0155 | 0.0584 | 0.0079 | 0.0329 | 0.0406 | 0 |
| Jaccard | AVG | 0.7466 | 0.7055 | 0.6994 | 0.9031 | 0.7413 | 0.7662 | 0.7811 | 0.6164 | 0.7297 | 0.9286 |
| | STD | 0.0311 | 0.0717 | 0.0648 | 0.0207 | 0.0274 | 0.0640 | 0.0196 | 0.0228 | 0.0405 | 0 |
| AUC | AVG | 0.8293 | 0.8418 | 0.8534 | 0.9623 | 0.8467 | 0.7438 | 0.9311 | 0.5863 | 0.8127 | 0.9643 |
| | STD | 0.0206 | 0.0417 | 0.0338 | 0.0086 | 0.0150 | 0.0545 | 0.0071 | 0.0262 | 0.0376 | 0 |

TABLE 16. Performance metrics of the BSSA algorithm with the UCI testing datasets over 20 independent runs.

| | | HeartEW | SonarEW | Ionosphere | WineW | Clean1 | Colon | Movement | German | Austurlian | Leukemia |
|----------------|-----|---------|---------|------------|--------|--------|---------------|----------|--------|------------|----------|
| Accuracy | AVG | 0.7562 | 0.7395 | 0.8264 | 0.9702 | 0.8324 | 0.9889 | 0.8790 | 0.6953 | 0.7951 | 0.9310 |
| | STD | 0.0426 | 0.0283 | 0.0258 | 0.0253 | 0.0215 | 0.0228 | 0.0142 | 0.0144 | 0.0309 | 0.0288 |
| F-measure | AVG | 0.7982 | 0.6759 | 0.6812 | 0.9713 | 0.8344 | 0.9918 | 0.8779 | 0.7964 | 0.8230 | 0.9476 |
| | STD | 0.0329 | 0.0404 | 0.0612 | 0.0250 | 0.0233 | 0.0168 | 0.0144 | 0.0107 | 0.0282 | 0.0211 |
| Precision | AVG | 0.7443 | 0.8071 | 0.9720 | 0.9709 | 0.9391 | 0.9885 | 0.8911 | 0.7483 | 0.7885 | 0.9643 |
| | STD | 0.0554 | 0.0389 | 0.0266 | 0.0239 | 0.0164 | 0.0282 | 0.0144 | 0.0113 | 0.0292 | 0.0366 |
| Recall | AVG | 0.8678 | 0.5828 | 0.5270 | 0.9729 | 0.7513 | 0.9958 | 0.8790 | 0.8514 | 0.8623 | 0.9321 |
| | STD | 0.0655 | 0.0486 | 0.0688 | 0.0256 | 0.0327 | 0.0186 | 0.0142 | 0.0222 | 0.0473 | 0.0160 |
| Specificity | AVG | 0.6167 | 0.8773 | 0.9918 | 0.9850 | 0.9371 | 0.9750 | 0.9914 | 0.3311 | 0.7120 | 0.9286 |
| | STD | 0.1177 | 0.0269 | 0.0076 | 0.0133 | 0.0173 | 0.0611 | 0.0010 | 0.0462 | 0.0504 | 0.0733 |
| Geometric Mean | AVG | 0.7265 | 0.7144 | 0.7215 | 0.9787 | 0.8389 | 0.9848 | 0.9310 | 0.5294 | 0.7826 | 0.9297 |
| | STD | 0.0567 | 0.0329 | 0.0484 | 0.0196 | 0.0215 | 0.0324 | 0.0084 | 0.0340 | 0.0315 | 0.0393 |
| Jaccard | AVG | 0.6654 | 0.5118 | 0.5196 | 0.9456 | 0.7165 | 0.9843 | 0.7933 | 0.6617 | 0.7001 | 0.9012 |
| | STD | 0.0476 | 0.0465 | 0.0690 | 0.0460 | 0.0345 | 0.0323 | 0.0204 | 0.0149 | 0.0407 | 0.0387 |
| AUC | AVG | 0.8326 | 0.8136 | 0.9094 | 0.9652 | 0.9371 | 0.9972 | 0.9352 | 0.6634 | 0.8555 | 0.9926 |
| | STD | 0.0342 | 0.0217 | 0.0173 | 0.0123 | 0.0087 | 0.0061 | 0.0076 | 0.0259 | 0.0353 | 0.0031 |

TABLE 17. Performance metrics of the IBSSA algorithm with the UCI testing datasets over 20 independent runs.

| | | HeartEW | SonarEW | Ionosphere | WineW | Clean1 | Colon | Movement | German | Austurlian | Leukemia |
|----------------|-----|---------|---------|------------|--------|--------|--------|----------|--------|------------|----------|
| Accuracy | AVG | 0.8525 | 0.7516 | 0.8356 | 0.9625 | 0.8296 | 0.9472 | 0.8757 | 0.6970 | 0.8296 | 0.9310 |
| | STD | 0.0290 | 0.0368 | 0.0192 | 0.0327 | 0.0176 | 0.0525 | 0.0156 | 0.0147 | 0.0303 | 0.0243 |
| F-measure | AVG | 0.8712 | 0.6908 | 0.7038 | 0.9640 | 0.8336 | 0.9594 | 0.8740 | 0.7967 | 0.8519 | 0.9475 |
| | STD | 0.0268 | 0.0487 | 0.0435 | 0.0328 | 0.0190 | 0.0409 | 0.0155 | 0.0113 | 0.0235 | 0.0176 |
| Precision | AVG | 0.8445 | 0.8257 | 0.9745 | 0.9624 | 0.9256 | 0.9714 | 0.8893 | 0.7508 | 0.8253 | 0.9679 |
| | STD | 0.0203 | 0.0514 | 0.0269 | 0.0338 | 0.0177 | 0.0401 | 0.0154 | 0.0074 | 0.0466 | 0.0365 |
| Recall | AVG | 0.9000 | 0.5948 | 0.5527 | 0.9675 | 0.7587 | 0.9500 | 0.8757 | 0.8488 | 0.8833 | 0.9286 |
| | STD | 0.0378 | 0.0523 | 0.0537 | 0.0304 | 0.0284 | 0.0628 | 0.0156 | 0.0189 | 0.0377 | 0 |
| Specificity | AVG | 0.7931 | 0.8894 | 0.9918 | 0.9815 | 0.9210 | 0.9417 | 0.9911 | 0.3428 | 0.7630 | 0.9357 |
| | STD | 0.0262 | 0.0344 | 0.0090 | 0.0162 | 0.0202 | 0.0816 | 0.0011 | 0.0211 | 0.0780 | 0.0729 |
| Geometric Mean | AVG | 0.8447 | 0.7268 | 0.7395 | 0.9742 | 0.8357 | 0.9446 | 0.9288 | 0.5391 | 0.8194 | 0.9315 |
| | STD | 0.0280 | 0.0404 | 0.0353 | 0.0235 | 0.0175 | 0.0550 | 0.0091 | 0.0176 | 0.0365 | 0.0365 |
| Jaccard | AVG | 0.7727 | 0.5297 | 0.5446 | 0.9327 | 0.7151 | 0.9247 | 0.7872 | 0.6623 | 0.7427 | 0.9007 |
| | STD | 0.0412 | 0.0594 | 0.0520 | 0.0581 | 0.0281 | 0.0743 | 0.0228 | 0.0154 | 0.0349 | 0.0316 |
| AUC | AVG | 0.9025 | 0.8236 | 0.9146 | 0.9630 | 0.9341 | 0.9771 | 0.9334 | 0.8873 | 0.8827 | 0.9921 |
| | STD | 0.0104 | 0.0319 | 0.0252 | 0.0058 | 0.0087 | 0.0248 | 0.0084 | 0.0131 | 0.0257 | 0.0035 |

TABLE 18. Comparison of the proposed system and other existing crack detection systems.

| Study | Crack | | Feature | Optimization | Results |
|-----------------|-----------|----------|---|--------------|---|
| | Detection | Severity | | | |
| Proposed System | ✓ | ✓ | HOG, LBP for crack detection - CNN-learned feature / LBP, HOG for crack severity | ✓ | Crack detection: Accuracy: 96.86% F-measure: 96.22% Crack severity recognition: Accuracy: 80.41% F-measure: 80.50% |
| SSA | | | | | |
| [31] 2019 | ✓ | × | CNN-learned features | × | Accuracy: — F-measure: 98.97% |
| [6] 2019 | ✓ | × | CNN-learned features, SURF features | × | Accuracy: 98% F-measure: 95% |
| [39] 2019 | ✓ | × | CNN-learned features based atrous convolution, ASPP module and depth-wise separable convolution | × | Accuracy: 96.37% F-measure: 87.71% |
| [35] 2018 | ✓ | × | CNN-learned features | × | Accuracy: — F-measure: 89.65% |
| [38] 2018 | ✓ | × | LBP, HOG | × | Accuracy: — F-measure: 91.03% |
| [32] 2018 | ✓ | × | CNN-learned features | × | Accuracy: 92.2% F-measure: — |
| [34] 2018 | ✓ | × | CNN-learned features | × | Full training mode: Accuracy: 87.54% F-measure: — Transfer learning mode: Accuracy: 89.31% F-measure: — |
| [33] 2018 | ✓ | × | CNN-learned features with edge detector | × | AlexNet, TL mode: Accuracy: 98% F-measure: — FT and classification modes: Accuracy: 97%, similarly. F-measure: — |
| [37] 2018 | ✓ | × | CNN-learned features | × | Accuracy: — F-measure: 89.59% |
| [36] 2017 | ✓ | × | CNN-learned features | × | Accuracy: 97.95% F-measure: — |

The average and standard deviation of the different performance metrics for the achieved results from performing the BSSA feature selection method on 10 UCI datasets over the 20 independent runs are illustrated in Table 16.

The average and standard deviation of the different performance metrics for the obtained results from performing the IBSSA feature selection method on 10 UCI datasets over the 20 independent runs are illustrated in Table 17.

D. THE TIME COMPLEXITY OF MO-IBSSA

The computational complexity of the proposed MO-IBSSA algorithm is influenced by the computation of the objective function, the crowding distance and the non-dominated comparison of the salps in the population and the archive. If there are M objective functions and N number of search agents (salps) in the population:

- The process of updating the archive can be of $O(M * N^2)$ in the worst case, as all salps in the population may wish to enter into the archive.
- The computation of the objective function has $O(Cof * N)$ computational complexity.
- The computational complexity of updating the Salps position is $O(d * N)$.

Thus, the overall complexity of the proposed MO-IBSSA is $O(t * (M * N^2 + Cof * N + d * N))$, where M indicates the number of objectives functions, t represents the number of iterations, d is the number of variables (dimension), N is the number of solutions, and Cof indicates the cost of objective function. Despite observing that the proposed MO-IBSSA approach achieved the same computational complexity of the basic multi-objective SSA version, it also achieved less computational time through using the filter-based pre-selection phase that reduces the feature set dimensionality.

E. COMPARATIVE ANALYSIS AGAINST STATE-OF-THE-ART CRACK DETECTION approaches

Table 18 shows a comparative summary for characteristics of the proposed approach against several state-of-the-art crack detection approaches.

From Table 18, it is noticed that the proposed system outperforms the model based on HOG and LBP proposed in [38], with F-measure ratio. The proposed system obtains F-measure ratio of 96.86% using hand-crafted features. Compared to the systems proposed in [6], [32], [34], [35], [37], [39] based on CNN deep learning models, the proposed system outperforms all of them in terms of accuracy. The proposed system achieves an accuracy of 96.86% for crack detection datasets. Moreover, the proposed system has a feature reduction rate of 64.86%-68.8% and the shortest computational time. On the other hand, the proposed system shows slightly lower accuracy by a maximum of 2.14% against all of the proposed systems in [31], [33], [36], but it used only approximately 33% of features with the shortest time, there is no need for high computational resources. However, as accuracy alone is not sufficient for reflecting the actual performance of crack detection systems, in this paper several additional performance metrics have been measured. Finally, up to our knowledge, the proposed system is the first system, which uses optimization techniques in the field of crack detection.

F. LIMITATIONS

As illustrated in the previous sections, there are a lot of advantages of the proposed system, such as having a feature reduction rate of approximately 61% on average for both crack detection and crack severity recognition, reducing computational time to approximately half of the original time, and improving the performance rate of crack detection. Moreover, as far as we know, it is considered the first trial for developing a system that handles the problem of crack severity recognition based on a hybrid filter-wrapper with multi-objective

bio-inspired optimization feature selection method. However, the proposed MO-IBSSA feature selection method has several observed limitations to be stated as well. The limitation of the proposed MO-IBSSA is that it still has a lower feature reduction rate, where it selects more features than other optimization algorithms over most of the used datasets. Therefore, a new filter-based selection strategy is used as a pre-selection phase to strengthen the proposed algorithm to select fewer features. Also, the proposed MO-IBSSA feature selection method has other limitations, as listed below. 1) the proposed crack severity recognition is still suffering high confusion between moderate and severe crack severity, 2) the proposed feature pre-selection step applies only filtering method aiming to select the high-ranked features without considering the redundant features, and 3) another limitation related to the fitness function, where the Kappa index may not be an accurate reflector of the true level of agreement between raters as it is influenced by the prevalence of the condition under observation.

VI. CONCLUSION AND FUTURE WORK

This paper proposed a novel crack detection and crack severity recognition system that utilizes a hybrid filter-wrapper feature selection method based on a Multi-Objective Improved Binary Salp Swarm Algorithm. The proposed system consists of image acquisition, data preparation through dividing crack images into small patches, data cleansing and augmentation, feature extraction, fusion, selection, and classification. In order to train and validate the proposed system, we implemented multiple experiments with 4 collected primary real-world datasets and other selected UCI publicly available benchmark datasets. Based on the obtained experimental results, the essential finding is that the proposed crack detection system in general improves the accuracy by $\approx 27\%$ - 57% and the F-measure by $\approx 23\%$ - 42% compared to the state-of-the-art classification using the whole set of features with an accuracy that ranges 93.46%-96.84% on both Crack and Intact classes and with high discriminatory power AUC that ranges 94.19%-98.42%. Moreover, a feature reduction rate of approximately 54.9% - 68.8% was achieved with all datasets.

Moreover, for crack severity recognition, it was observed that using the VGG16 CNN learned features outperformed the performance of the fused hand-crafted features by 17.7%, 15.9%, and 23.5% for fine, moderate, and severe cracks, respectively. Also, using the CNN learned features reduced the confusion between crack severity degrees. Moreover, it is worthy mentioning that the proposed system with CNN learned features led to a slight degradation in performance by only 0.44%, while reducing the features by approximately 54%.

For future research, several challenges and research directions could be considered, such as investigating the performance of end-to-end deep learning models using the proposed feature selection approach. Also, applying semantic segmentation to crack images as an additional pre-processing

step to improve the recognition rate of crack severity is another challenge.

REFERENCES

- [1] A. Perles, E. Pérez-Marín, R. Mercado, J. D. Segrelles, I. Blanquer, and M. Zarzo, and F. J. García-Diego, "An energy-efficient Internet of Things (IoT) architecture for preventive conservation of cultural heritage," *Future Gener. Comput. Syst.*, vol. 81, pp. 566–581, Apr. 2018.
- [2] S. Brown, I. Cole, V. Daniel, S. King, and C. Pearson, "Guidelines for environmental control in cultural institutions," Heritage Collections Council, Austral. Inst. Conservation Cultural Mater., Canberra, ACT, Australia, Tech. Rep., 2002.
- [3] A. M. H. Abulnour, "Protecting the Egyptian monuments: Fundamentals of proficiency," *Alexandria Eng. J.*, vol. 52, no. 4, pp. 779–785, 2013.
- [4] M. Mohtasham Khani, S. Vahidnia, L. Ghasemzadeh, Y. E. Ozturk, M. Yuvalaklioglu, S. Akin, and N. K. Ure, "Deep-learning-based crack detection with applications for the structural health monitoring of gas turbines," *Struct. Health Monitor.*, Nov. 2019, Art. no. 147592171988320.
- [5] M. Abdulkarem, K. Samsudin, F. Z. Rokhani, and M. F. A. Rasid, "Wireless sensor network for structural health monitoring: A contemporary review of technologies, challenges, and future direction," *Struct. Health Monitor.*, vol. 19, no. 3, pp. 693–735, 2020.
- [6] H. Kim, E. Ahn, M. Shin, and S.-H. Sim, "Crack and noncrack classification from concrete surface images using machine learning," *Struct. Health Monitor.*, vol. 18, no. 3, pp. 725–738, May 2019.
- [7] D. J. Atha and M. R. Jahanshahi, "Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection," *Struct. Health Monitor.*, vol. 17, no. 5, pp. 1110–1128, Sep. 2018.
- [8] S. Sadeg, L. Hamdad, K. Benatchba, and Z. & Habbas, "Bso-fs: Bee swarm optimization for feature selection in classification," in *Proc. 13th Int. Work-Conf. Artif. Neural Netw.*, Palma de Mallorca, Spain, vol. 9094, Jun. 2015, pp. 387–399.
- [9] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, Lake Tahoe, Nevada, Dec. 2012, pp. 2951–2959.
- [10] SA Taie and W. Ghonaim, "CSO-based algorithm with support vector machine for brain tumor's disease diagnosis," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PerCom Workshops)*, Kona, HI, USA, Mar. 2017, pp. 183–187.
- [11] M. Hammami, S. Bechikh, C.-C. Hung, and L. Ben Said, "A multi-objective hybrid filter-wrapper evolutionary approach for feature selection," *Memetic Comput.*, vol. 11, no. 2, pp. 193–208, Jun. 2019.
- [12] R. Hancer, B. "Xue, & M. Zhang, "Differential evolution for filter feature selection based on information theory and feature ranking," *Knowl.-Based Syst.*, vol. 140, pp. 103–119, Jan. 2018.
- [13] Y. Zhang, D.-W. Gong, X.-Z. Gao, T. Tian, and X.-Y. Sun, "Binary differential evolution with self-learning for multi-objective feature selection," *Inf. Sci.*, vol. 507, pp. 67–85, Jan. 2020.
- [14] R. A. Ibrahim, A. A. Ewees, D. Oliva, M. Abd Elaziz, and S. Lu, "Improved salp swarm algorithm based on particle swarm optimization for feature selection," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 8, pp. 3155–3169, Aug. 2019.
- [15] T. Kaur, B. S. Saini, and S. Gupta, "An optimal spectroscopic feature fusion strategy for MR brain tumor classification using Fisher criteria and parameter-free BAT optimization algorithm," *Biocybernetics Biomed. Eng.*, vol. 38, no. 2, pp. 409–424, 2018.
- [16] S. Sasikala, M. Ezhilarasi, and S. A. Kumar, "Detection of breast cancer using fusion of MLO and CC view features through a hybrid technique based on binary firefly algorithm and optimum-path forest classifier," in *Applied Nature-Inspired Computing: Algorithms and Case Studies* (Springer Tracts in Nature-Inspired Computing). 2020, pp. 23–40.
- [17] S. Gunasundari, S. Janakiraman, and S. Meenambal, "Velocity bounded Boolean particle swarm optimization for improved feature selection in liver and kidney disease diagnosis," *Expert Syst. Appl.*, vol. 56, pp. 28–47, Sep. 2016.
- [18] M. Alirezai, S. T. A. Niaki, and S. A. A. Niaki, "A bi-objective hybrid optimization algorithm to reduce noise and data dimension in diabetes diagnosis using support vector machines," *Expert Syst. Appl.*, vol. 127, pp. 47–57, Aug. 2019.
- [19] J. González, J. Ortega, M. Damas, P. Martín-Smith, and J. Q. Gan, "A new multi-objective wrapper method for feature selection—Accuracy and stability analysis for BCI," *Neurocomputing*, vol. 333, pp. 407–418, Mar. 2019.
- [20] I. Aljarah, M. Mafarja, A. A. Heidari, H. Faris, Y. Zhang, and S. Mirjalili, "Asynchronous accelerating multi-leader salp chains for feature selection," *Appl. Soft Comput.*, vol. 71, pp. 964–979, Oct. 2018.
- [21] H. Faris, M. M. Mafarja, A. A. Heidari, I. Aljarah, A. M. Al-Zoubi, S. Mirjalili, and H. Fujita, "An efficient binary Salp Swarm Algorithm with crossover scheme for feature selection problems," *Knowl.-Based Syst.*, vol. 154, pp. 43–67, Aug. 2018.
- [22] A. E. Hegazy, M. A. Makhlof, and G. S. El-Tawel, "Improved salp swarm algorithm for feature selection," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 32, no. 3, pp. 335–344, 2020.
- [23] A. E. Hegazy, M. A. Makhlof, and G. S. El-Tawel, "Feature selection using chaotic salp swarm algorithm for data classification," *Arabian J. Sci. Eng.*, vol. 44, no. 4, pp. 3801–3816, 2018.
- [24] H. M. Zawbaa, E. Emary, C. Grosan, and V. Snasel, "Large-dimensionality small-instance set feature selection: A hybrid bio-inspired heuristic approach," *Swarm Evol. Comput.*, vol. 42, pp. 29–42, Oct. 2018.
- [25] H. Chantari, M. Mafarja, H. Alsawalqah, A. A. Heidari, I. Aljarah, and H. Faris, "Feature selection using binary grey wolf optimizer with elite-based crossover for arabic text classification," *Neural Comput. Appl.*, pp. 1–20, Jul. 2019.
- [26] M. Abdel-Basset, D. El-Shahat, I. El-Henawy, V. H. C. De Albuquerque, and S. Mirjalili, "A new fusion of grey wolf optimizer algorithm with a two-phase mutation for feature selection," *Expert Syst. Appl.*, vol. 139, Jan. 2020, Art. no. 112824.
- [27] M. Mafarja and S. Mirjalili, "Whale optimization approaches for wrapper feature selection," *Appl. Soft Comput.*, vol. 62, pp. 441–453, Jan. 2018.
- [28] T. Thaher, A. A. Heidari, M. Mafarja, J. S. Dong, and S. Mirjalili, "Binary Harris Hawks Optimizer for high-dimensional, low sample size feature selection," in *Evolutionary Machine Learning Techniques* (Algorithms for Intelligent Systems). Singapore: Springer, 2019, pp. 251–272.
- [29] J. Too, A. Abdullah, and M. Saad, "A new quadratic binary Harris hawk optimization for feature selection," *Electronics*, vol. 8, no. 10, p. 1130, 2019.
- [30] R. Sindhu, R. Ngadiran, Y. M. Yacob, N. A. H. Zahri, and M. Hariharan, "Sine-cosine algorithm for feature selection with elitism strategy and new updating mechanism," *Neural Comput. Appl.*, vol. 28, no. 10, pp. 2947–2958, 2017.
- [31] K. Jang, N. Kim, and Y.-K. An, "Deep learning-based autonomous concrete crack evaluation through hybrid image scanning," *Struct. Health Monitor.*, vol. 18, nos. 5–6, pp. 1722–1737, Nov. 2019.
- [32] W. R. L. D. Silva and D. S. D. Lucena, "Concrete cracks detection based on deep learning image classification," *Proceedings*, vol. 2, no. 8, p. 489, 2018.
- [33] S. Dorafshan, R. J. Thomas, and M. Maguire, "Comparison of deep convolutional neural networks and edge detectors for image-based crack detection in concrete," *Construct. Building Mater.*, vol. 186, pp. 1031–1045, Oct. 2018.
- [34] S. Dorafshan, R. J. Thomas, and M. Maguire, "SDNET2018: An annotated image dataset for non-contact concrete crack detection using deep convolutional neural networks," *Data Brief*, vol. 21, pp. 1664–1668, Dec. 2018.
- [35] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiya, 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, Dec. 2018.
- [36] Y. J. Cha, W. Choi, and O. Büyükoztürk, "Deep learning-based crack damage detection using convolutional neural networks," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 32, no. 5, pp. 361–378, 2017.
- [37] K. Zhang, H. D. Cheng, and B. Zhang, "Unified approach to pavement crack and sealed crack detection using preclassification based on transfer learning," *J. Comput. Civil Eng.*, vol. 32, no. 2, Mar. 2018, Art. no. 04018001.
- [38] B. Wang, W. Zhao, P. Gao, Y. Zhang, and Z. Wang, "Crack damage detection method via multiple visual features and efficient multi-task learning model," *Sensors*, vol. 18, no. 6, p. 1796, 2018.
- [39] H. Xu, X. Su, Y. Wang, H. Cai, K. Cui, and X. Chen, "Automatic bridge crack detection using a convolutional neural network," *Appl. Sci.*, vol. 9, no. 14, p. 2867, 2019.
- [40] B. Li, K. Cheng, and Z. Yu, "Histogram of oriented gradient based gist feature for building recognition," *Comput. Intell. Neurosci.*, vol. 2016, pp. 1–9, Oct. 2016.
- [41] S. Azam and M. Gavrilova, "License plate image patch filtering using HOG descriptor and bio-inspired optimization," in *Proc. Comput. Graph. Int. Conf. (CGI)*, New York, NY, USA, 2017, pp. 1–6.

- [42] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiwayama, and H. Omata, "Road damage detection and classification using deep neural networks with smart-phone images," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 33, no. 12, pp. 1127–1141, Dec. 2018.
- [43] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiolog," *Insights into Imag.*, vol. 9, no. 9, pp. 611–629, 2018.
- [44] D. Scherer, A. Muller, and S. Behnke, "Evaluation of pooling operations in convolutional architectures for object recognition," in *Proc. Int. Conf. Artif. Neural Netw. (ICANN)* (Lecture Notes in Computer Science), 2010, pp. 92–101.
- [45] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, "Towards better analysis of deep convolutional neural networks," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 91–100, Jan. 2017.
- [46] B. C. Ross, "Mutual information between discrete and continuous data sets," *PLoS ONE*, vol. 9, no. 2, 2014, Art. no. e87357.
- [47] Z. Chen, X. Han, C. Fan, T. Zheng, and S. Mei, "A two-stage feature selection method for power system transient stability status prediction," *Energies*, vol. 12, no. 4, p. 689, 2019.
- [48] N. El-Bendary, E. E. Hariri, A. E. Hassanien, and A. Badr, "Using machine learning techniques for evaluating tomato ripeness," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 1892–1905, 2015.
- [49] B. Ghaddar and J. Naoum-Sawaya, "High dimensional data classification and feature selection using support vector machines," *Eur. J. Oper. Res.*, vol. 265, no. 3, pp. 993–1004, Mar. 2018.
- [50] A. Mikolajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *Proc. Int. Interdiscipl. PhD Workshop (IIPhDW)*, Swinoujscie, Poland, May 2018, pp. 117–122.
- [51] S. Mirjalili, A. H. Gandomi, S. Z. Mirjalili, S. Saremi, H. Faris, and S. M. Mirjalili, "Salp swarm algorithm: A bio-inspired optimizer for engineering design problems," *Adv. Eng. Softw.*, vol. 114, pp. 163–191, Dec. 2017.
- [52] *UCI Machine Learning Repository*. Accessed: Dec. 10, 2019. [Online]. Available: <https://archive.ics.uci.edu/ml/index.php>



is currently a Lecturer Assistant with the Faculty of Computers and Information, Fayoum University. She is also the Deputy Director of Alumni Follow-Up Unit. She has published several articles in international journals and peer-reviewed international conference proceedings along with a book chapter. Her research interests include machine learning, image and signal processing, bio-inspired optimization, wireless sensors network cloud computing security, and the IoT.

ESRAA ELHARIRI received the B.Sc. degree (Hons.) in computer science from the Faculty of Computers and Information (FCI), Fayoum University, Egypt, in 2010, and the M.Sc. degree from FCI-Cairo University, Egypt, in 2015. Her M.Sc. topic was about content-based image retrieval for agricultural crops. She is currently pursuing the Ph.D. degree with FCI-Fayoum University. While her Ph.D. topic is about IOT-based technique for historical and cultural heritage preservation. She



NASHWA EL-BENDARY (Senior Member, IEEE) received the Ph.D. degree in information technology from the Faculty of Computers and Information, Cairo University, Egypt, in 2008. From 2008 to 2015, she was an Assistant Professor with the College of Management and Technology. Since 2016, she has been an Associate Professor with the College of Computing and Information Technology, Arab Academy for Science, Technology, and Maritime Transport (AASTMT). She has authored many articles published in indexed international journals and conferences. She also participated in several international research projects. Her research interests include image and signal processing, machine learning, pattern recognition, and the Internet of Things.

Dr. El-Bendary was a recipient of the UNESCO-ALECSO Award for Creativity and Technical Innovation for Young Researchers, in 2014, and The L'Oréal-UNESCO For Women in Science Fellowship, in 2015.



SHEREEN A. TAIE received the B.Sc. degree in computer science from the Faculty of Science, Cairo University, in 1996, and the M.S. and Ph.D. degrees in computer science from the Computer and Mathematics Department, Faculty of Science, Cairo University, in 2006 and 2012, respectively. She is currently the Vice Dean for Students Affairs, Faculty of Computer and Information, Fayoum University, Egypt, where she is also an Associate Professor with the Computer Science Department,

Faculty of Computers and Information, and the Head of Center of Electronic Courses Production.

...