

Performance Analysis of Improved Swarm Intelligence Based Classifier for Fabric Defect Detection

V. GNANAPRAKASH AND PONNUSAMY THANGAPANDIAN VANATHI²

¹*Department of Electronics and Communication Engineering
Bannari Amman Institute of Technology
Tamil Nadu, 638401 India*

²*Department of Electronics and Communication Engineering
PSG College of Technology
Tamil Nadu, 641004 India
E-mail: gnanam1987@gmail.com*

Automatic defect detection in fabrics is one of the most essential systems used in the textile industry to check the quality of the fabric. In most of the existing systems, a learning-based approach is implemented for defect detection in simple patterned fabrics. In this paper, swarm intelligence-based Backpropagation Neural Network (BPNN) classifiers are implemented for defect detection in complex patterned fabrics. But the problem with existing Binary Particle Swarm Optimization (BPSO) based BPNN classifier is premature convergence. To offset this problem an evolutionary state-based greedy reset is proposed to promote an effective and efficient search of the particles in the search space of the BPSO algorithm. The proposed system comprises of feature extraction phase followed by a performance evaluation phase. The combinations of features namely (i) Gray Level Co-occurrence Matrix (GLCM); (ii) Discrete Wavelet Transform (DWT) and GLCM (W-GLCM); (iii) DWT, Local Binary Pattern (LBP), and GLCM (WL-GLCM) are extracted from the complex patterned fabrics and their performances are evaluated by employing swarm intelligence-based Backpropagation Neural Network (BPNN) classifier. The proposed system is validated with fabric datasets taken from the TILDA fabric database. From the results, it is observed that proposed system classification accuracy is 99.75% and it is better than the existing work with 77% reduced features.

Keywords: gabor filter, discrete wavelet transform, local binary pattern, binary particle swarm optimization, backpropagation neural networks

1. INTRODUCTION

Defect detection in the fabric is essential for quality control in textile products. In defective fabric, the weave pattern of the fabric may differ from the original design due to the wrong mechanical movement or breakage of thread on a loom. Due to this, defective fabric selling price may drop by 45%-65%. Human experts are utilized for traditional inspection of fabrics. However, the performance and reliability of the traditional inspection techniques are low. Hence automation is necessary to inspect the fabric quality in the textile industry.

Fabric characteristics are based on the texture and each fabric has a different texture namely uniform, random [1-6] and patterned texture [7, 8]. Due to the varying characteristics of the fabric, developing a fabric defect detection system with good accuracy is still a challenging task. The fabrics can be grouped into 17 categories namely p1, p2, p3, p3m1,

Received April 8, 2020; revised July 26 & September 11, 2020; accepted October 28, 2020.
Communicated by Shyi-Ming Chen.

p31m, p4, p4m, p4g, pm, pg, pmg, pgg, p6, p6m, cm, cmm, and pmm based on the unit shape and its symmetry [9]. In the p1 group of fabric, the basic unit is repeated over the entire fabrics like plain weave fabric and in this type of fabric, pattern translation is allowed to get the texture. The non-p1 groups of fabrics involve rotation, reflection and glide reflections to get the texture.

Many systems have been developed in the past years [41, 42] for defect detection in fabrics. Most of these methods are particularly applied for simple patterned fabrics. There are only a few systems which detect faults in complex patterned fabrics.

A Gabor filter with two scales and six orientations was introduced for fabric inspection with a satisfying performance by the authors L. H. Ding *et al.*, and Junfeng Jing *et al.* [10, 11]. Multiple adaptive wavelets have been used to enhance the performance of the fabric defect inspection system proposed by X. Yang, G. Pang and N. Yung [12] and the clustering approach was used by Gang Yu, Sagar V. Kamarthi to extract features from the discrete wavelet transform coefficients [13].

Defects in fabrics are detected by LBP based texture features [14] and a two-dimensional wavelet transform is used in the early step to speed up the system running time. GLCM based co-occurrence features are extracted from the textured fabrics and finally, classification is done using a learning vector quantization based neural network to achieve better performance [15]. Kernel-based fuzzy *c*-means clustering (KFCM) technique has been used for weave pattern classification using the texture features such as mean and variance [16]. Spectral, model-based, and statistical feature extraction techniques were used for defect detection in fabric [17] and comparisons were done for defect detection of fabrics using GLCM and Gabor Filter [18]. Fabric defect detection systems were implemented based on GLCM features and BPNN classifier [44-46]. Optimum weight and threshold values of BPNN are identified using particle swarm optimization (PSO) for defect classification in fabrics [47]. The optimal parameters of the Elliptical Gabor filter are identified by employing random drift PSO for defect detection in fabrics [48]. Fabric inspection system is carried out using a defect segmentation algorithm followed by an artificial neural network [49]. Textile defect detection and classification algorithms are developed by employing statistical feature extraction followed by a neural network classifier [50]. Textile defect detection system is implemented with multi-layer neural networks to classify the textile defects and detect the defects with a microcontroller-based mechanical system [51]. Most of the above-mentioned literature deals with simple patterned fabrics.

In this paper, a novel defect detection technique is proposed to cater to the complex patterned fabrics. It is based on analyzing texture patterns by extracting effective statistical features from the combination of DWT, LBP, and GLCM for defect detection in fabrics. Among various combinations, features extracted from the combination of DWT, LBP and GLCM are more appropriate for describing the local texture information of the complex patterned fabric. The effectiveness of the extracted features is assessed by the evolutionary state greedy reset BPSO based BPNN classifier. It overcomes the problem of premature convergence in the standard BPSO based classifier by introducing unique self-adaptive control parameters for each particle that is controlled by the evolutionary state. Therefore, proposed classifier can discriminate the features of defective fabric and defect-free fabric with improved performance than the existing methods.

The rest of this paper is organized as follows: Section 2 presents the proposed methodology for co-occurrence feature extraction and feature evaluation using swarm intelli-

gence-based classifiers. Section 3 presents the experimental results on the TILDA image database to validate the proposed method.

2. PROPOSED WORK

In this paper, a hybrid version of the feature extraction technique is introduced to cater to the complex patterned fabric with improved accuracy. The proposed method flow diagram is shown in Fig. 1. The detailed description of the various modules present in the flow diagram is discussed below.

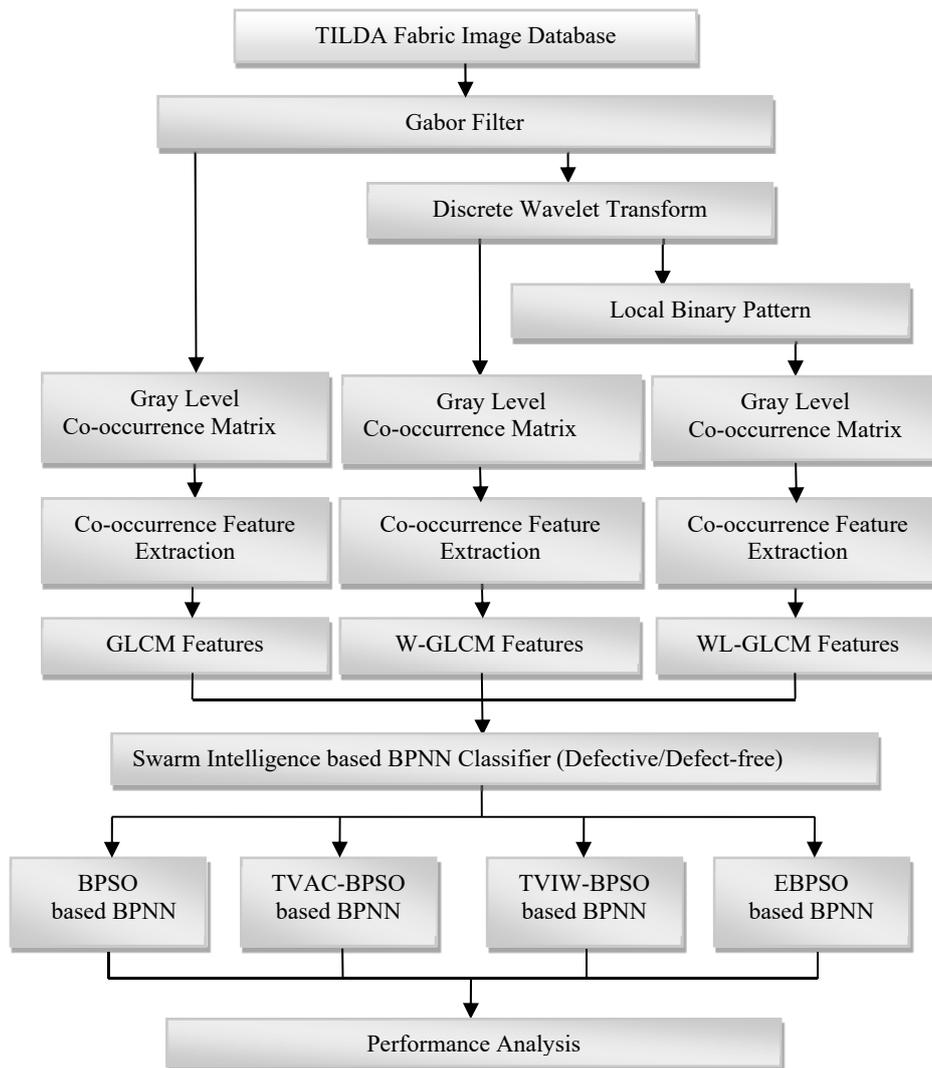


Fig. 1. Proposed method flow diagram.

2.1 Gabor Filter

Gabor filters resemble the performance of the human visual system for extracting features at different orientations and scales. This multi-resolution capability of Gabor filters may be useful for extracting significant information from the input texture image. The mathematical form of Gabor filter [24] with the center frequency f_i , orientation θ_k modulated by a Gaussian envelope with standard deviations σ_x and σ_y is given in Eq. (1)

$$h(x, y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left\{-\frac{1}{2}\left[\frac{x^2}{\sigma_x^2} + \frac{y^2}{\sigma_y^2}\right]\right\} \cos(2\pi f_i x + \phi) \quad (1)$$

where $x = x\cos\theta + y\sin\theta$, $y = -x\sin\theta + y\cos\theta$, ϕ is a phase component and θ is the orientation which is calculated as $\theta = \frac{k\pi}{n}$, $k = \{1, 2, \dots, n\}$.

In this work pre-processing is done with Gabor filter to achieve optimal localization of the texture in spatial and frequency domain [23]. This results in a good description of the texture without high-frequency noise by which it improves the quality of the features in the proposed system.

2.2 Discrete Wavelet Transform (DWT)

In the second stage of the feature extraction process, DWT [19] is applied to analyze the input fabric images at a multi-resolution level. DWT produces a hierarchy based sub-band structure that decomposes the signal into four sub-bands namely LL1, LH1, HL1, HH1 bands in level 1 [25]. In this, the LL1 sub-band contains the approximation image with low-frequency components and the remaining three sub-bands contain detailed images with a high-frequency component. In level 2, the LL1 (low frequency) band is again decomposed into four more sub-bands by retaining other sub-bands and this process will be continued until to reach the desired low-frequency region. In the wavelet decomposition method, most of the energy is concentrated in the LL band (low frequency). Due to this property of DWT, the LL band alone is considered for the feature extraction process. The decomposition process of DWT is depicted in Fig. 2.

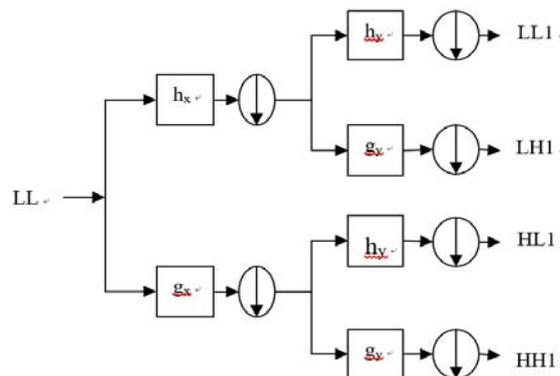


Fig. 2. Decomposition of discrete wavelet transform.

Low frequency and high-frequency selective filters are represented by the symbol h and g respectively, downsampling operation is represented by a down arrow inside a circle with a downsampling rate of 2. The localized texture features extracted from these sub-bands have unique characteristics that are more suitable for texture classification. In this work, Haar wavelet with level1 decomposition is used for the analysis of the textile images [26].

2.3 Local Binary Pattern (LBP)

In the next stage of the feature extraction process, local textural information is described by LBP. The LBP is a non-parametric method [20] to describe local texture by calculating the intensity difference between the center pixel and its neighbors. If its difference is positive, then binary one is assigned to the corresponding neighboring pixel intensity and if it is negative binary zero is assigned to the corresponding neighboring pixels. Therefore, a binary sequence is generated for all the pixel intensities in an image and finally, the center pixel intensity is replaced by a resultant decimal value obtained from the binary sequence. The entire process of LBP generation is illustrated in Fig. 3.

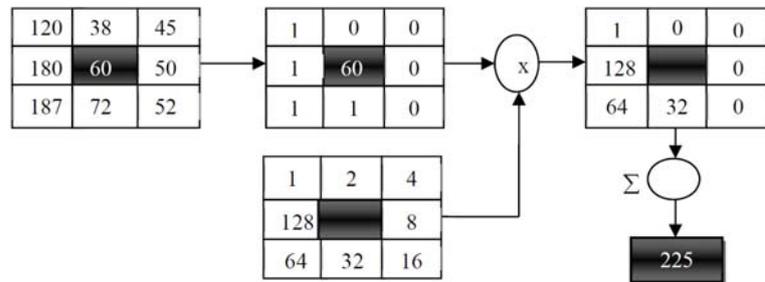


Fig. 3. LBP computation process.

Then the LBP histogram is calculated based on the obtained binary patterns from an image. The number of histogram bins required to represent an image will be decided by the number of neighboring pixels involved in LBP calculation. If it uses 8 neighboring pixels then the number of histogram bins required is 256 (*i.e.*, 2^8). LBP of an image is calculated using Eq. (2).

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} s(i_p - i_c) 2^p \tag{2}$$

where P is the number of local neighboring pixels, i_c is the gray intensity value of the center pixel of the local neighborhood, i_p is the gray intensity value of the neighboring pixels, and (x_c, y_c) are the coordinates of the center pixel of the local neighborhood. The function $s(x)$ is obtained from Eq. (3)

$$s(t) = \begin{cases} 0, & \text{if } t < 0 \\ 1, & \text{if } t \geq 0 \end{cases} \tag{3}$$

The calculated LBP image histogram describes the pattern distribution of the whole image.

2.4 Gray Level Co-occurrence Matrix (GLCM)

In the final stage of the feature extraction process, GLCM based co-occurrence features [21] are calculated to obtain the local texture information of the fabric. GLCM is one of the most common and extensively used techniques for texture measures. Image texture is one of the characters which help to identify an object. Gray level spatial dependency-based textural features are more significant in texture image classification. GLCM is a two-dimensional histogram of gray intensity levels in which it describes how often particular pair of pixels co-occur in an image with particular distance D and orientation angle θ . But the choice of distance and angle depends on the input database. In a texture image, any pixel is most likely to be correlated with the closely located neighbors than the neighbors which are located far away. Thus, the better performance of the classifier is achieved using GLCM with $D=1$ and 2. However, the co-occurring pixel pair is selected based on the angle θ . Possible θ values are 0° , 45° , 90° , and 135° . Different offset values are obtained by combining both the distance and angle which affects the performance of the matrix. In most of the early studies, horizontal offset [18] values have been used for the analysis. In this work, GLCM is generated with the offset value of the distance $D=1$ and angle $\theta=0^\circ$ [0 1]. The direct analysis of the GLCM is illustrated in Fig. 4 and the sample GLCM calculation is shown in Fig. 5 for the offset value of [0 1]. From the generated GLCM Haralick's defined second-order statistical (co-occurrence) features which are listed in Table 1 are calculated to measure the characteristics of the texture image.

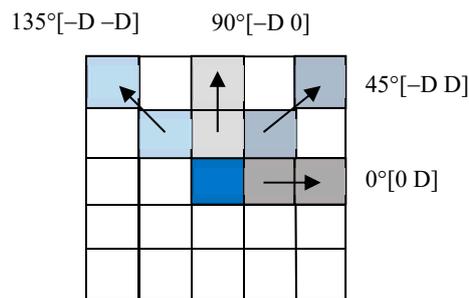


Fig. 4. GLCM direction analysis.

Generation of Gray Level Co-occurrence matrix with angle $\theta=0^\circ$ (Horizontal direction) and distance $D=1$ is illustrated in Fig. 5. In GLCM output, element (1,2) contains the value 1 because horizontally located adjacent pixel pairs 1 and 2 co-occurs only once in an image. Element (2,3) contains the value 2 because horizontally located adjacent pixel pairs 2 and 3 co-occurs twice in an image. Element (1,1) in GLCM has the value 0 because there are no horizontally located adjacent pixel pairs 1 and 1 in an image. In the same way remaining entries are calculated to obtain complete GLCM output.

The normalized co-occurrence matrices can be given as:

$$P(x, y) = \frac{G(x, y)}{\sum_{x=0}^{M-1} \sum_{y=0}^{M-1} G(x, y)} \quad (4)$$

where $G(x, y)$ is the number of occurrences of the gray levels g_x and g_y . Table 1 shows the Co-occurrence features that are extracted from the GLCM and its description.

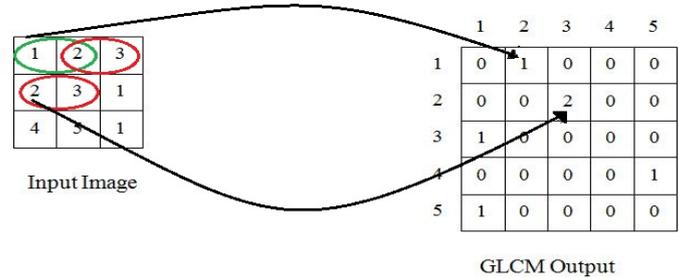


Fig. 5. Generation of GLCM for $\theta = 0^\circ$ and $D = 1$.

Table 1. Co-occurrence features.

Co-occurrence Features	Description
Contrast: $\sum_{x=0}^{M-1} \sum_{y=0}^{M-1} (x - y)^2 P(x, y)$	Measures local intensity variation present in an image
Correlation (M): $\frac{\sum_{x=0}^{M-1} \sum_{y=0}^{M-1} (x \times y) P(x, y) - \{\mu_x - \mu_y\}}{\sigma_x \sigma_y}$	Measures the gray level linear dependency between the pair of pixels.
Correlation: $\sum_{x=0}^{M-1} \sum_{y=0}^{M-1} P(x, y) \frac{(1 - \mu_x)(1 - \mu_y)}{\sigma_x \sigma_y}$	Measures the gray level linear dependency between the pair of pixels.
Cluster prominence: $\sum_{x=0}^{M-1} \sum_{y=0}^{M-1} (x + y - \mu_x - \mu_y)^4 P(x, y)$	Measures asymmetry of the GLCM
Cluster shade: $\sum_{x=0}^{M-1} \sum_{y=0}^{M-1} (x + y - \mu_x - \mu_y)^3 P(x, y)$	Measures uniformity of the GLCM
Dissimilarity: $\sum_{x=0}^{M-1} \sum_{y=0}^{M-1} (x - y) P(x, y)$	Measures the distance between pairs of pixels
Angular Second Moment (Energy): $\sum_{x=0}^{M-1} \sum_{y=0}^{M-1} \{P(x, y)\}^2$	Measures texture uniformity in an image to detect texture disorder.
Entropy: $\sum_{x=0}^{M-1} \sum_{y=0}^{M-1} P(x, y) \log_2(P(x, y))$	Measures degree of randomness in an image
Homogeneity (M): $\sum_{x=0}^{M-1} \sum_{y=0}^{M-1} \frac{P(x, y)}{1 + x - y }$	Measures the proximity between the distributed elements and the diagonal value of the GLCM.

Homogeneity: $\sum_{x=0}^{M-1} \sum_{y=0}^{M-1} \frac{P(x, y)}{1 + x - y ^2}$	Measures the proximity between the distributed elements and the diagonal value of the GLCM.
Maximum Probability: $\sum_{x=0}^{M-1} \sum_{y=0}^{M-1} \max(\max(P(x, y)))$	Measures the occurrences of the most predominant pair of neighboring pixels.
Variance: $\sum_{x=0}^{M-1} \sum_{y=0}^{M-1} (1 - \mu)^2 P(x, y)$	Measures the heterogeneity of the image
Sum Average: $\sum_{x=0}^{2M-1} x P_{i+j}(x)$	Measures the relationship between the pairs of pixels with lower intensity and the pairs of pixels with higher intensity.
Sum Variance: $\sum_{x=0}^{M-1} \{x - \sum_{x=0}^{2M-1} P_{i+j}(x) \log_2(P_{i+j}(x))\}^2 P_{i+j}(x)$	Measures average of heterogeneity in an image
Sum Entropy: $-\sum_{x=0}^{2M-1} P_{i+j}(x) \log_2(P_{i+j}(x))$	Measures degree of randomness in an image based on the probability distribution $P_{i+j}(x)$
Difference variance: $\sum_{x=0}^{M-1} \sum_{y=0}^{M-1} (x^2) P_{i-j}(x)$	Measures the heterogeneity of the image based on the probability distribution $P_{i-j}(x)$
Difference Entropy: $-\sum_{x=0}^{M-1} P_{i-j}(x) \log_2(P_{i-j}(x))$	Measures degree of randomness in neighborhoods intensity differences
Information Correlation 1: $\frac{H(XY) - H(XY1)}{H(X)}$	Measures the dependency between the probability distributions of x and y using mutual information.
Information Correlation 2: $\sqrt{1 - \exp(-2(H(XY2) - H(XY)))}$	Measures the dependency between the probability distributions of x and y .
Inverse Difference normalized: $\sum_{x=0}^{M-1} \sum_{y=0}^{M-1} \frac{P(x, y)}{1 + x - y /M}$	Measures local homogeneity of an image
Normalized Inverse Difference Moment: $\sum_{x=0}^{M-1} \sum_{y=0}^{M-1} \frac{P(x, y)}{1 + x - y ^2 / M^2}$	Measures local homogeneity of an image
Autocorrelation: $\sum_{x=0}^{M-1} \sum_{y=0}^{M-1} (x \times y) P(x, y)$	Measures the magnitude of the fineness and coarseness of texture

2.5 Swarm Intelligence Based Classifier

The performance of the extracted features is evaluated with the help of a swarm intelligence-based classifier by selecting the optimal features subset among all the features. The detailed description of the swarm intelligence-based classifiers present in the flow diagram is discussed below.

2.5.1 Binary Particle Swarm Optimization based classifier

Eberhart and Kennedy [27] introduced a population-based stochastic search algorithm, which derives evolutionary inspiration from lively movements and interactions of a group of birds and groups of fishes in the direction of searching for the food. In BPSO, the position of the particle represents a feasible solution in search space. A possible solution may be either global (best solution so far) or local (best solution among neighborhood particles), depending on the particle's position in the swarm. Particles in PSO moves with a particular velocity in a search space to get the optimal solution. The Memory element in every particle helps to remember its previous best solution. The position and velocity of the m^{th} particle in the n^{th} iteration are represented by $x^m(n)$ and $v^m(n)$ respectively. Personal best, $x_p^m(n)$ and global best, $x_G(n)$ are the two distinguished positions of each particle in PSO. In searching for food, the particle's position is updated based on its personal and global best positions. The velocity of the particle is tuned based on the corresponding particles and its neighbor's past behavior. In the standard PSO, particle's positions are initialized randomly with uniform distribution; whereas the particle's velocities are set to zero and all the particles are searching for its best solution in several iterations. In every n^{th} iteration, the velocity and positions of each particle are updated using Eqs. (5) and (6).

$$v^m(n+1) = w \times v^m(n) + c_1 \times \text{rand}_1(n)(x_p^m(n) - x^m(n)) + c_2 \times \text{rand}_2(n)(x_G(n) - x^m(n)); \quad (5)$$

$m = 1 \text{ to } N$

$$x^m(n+1) = x^m(n) + v^m(n); \quad m = 1 \text{ to } N \quad (6)$$

The parameters which influence for updating the velocity of each particle are the inertia weight (w), acceleration parameter namely cognitive parameter (c_1), social parameter (c_2), uniformly generated random values ($0 < \text{rand}_1 < 1$ and $0 < \text{rand}_2 < 1$) at each iteration, and population size (N). In this work, the constriction factor χ is used as an inertia weight w . This is shown in Eq. (7)

$$\chi = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|}, \quad (7)$$

where $\varphi = c_1 + c_2 > 4$. When Clerc's constriction factor is used, φ value is set to 4.1. For better convergence, cognitive parameter c_1 is set to 2.05 and the social parameter c_2 is set to 2.05.

In BPSO, particle positions are updated using Eq. (8). Based on Eq. (8), binary 1 is assigned if the particles get selected for the next iteration and binary 0 is assigned if they are not selected for the next iteration.

$$x^m(n+1) = \begin{cases} 1 & \text{if } rand < s^m(n), \\ 0 & \text{if otherwise} \end{cases} \quad (8)$$

where $s^m(n) = \frac{1}{1 + e^{-v^m(n)}}$ and the *rand* is the uniformly generated random values between 0 and 1.

The fitness function to be optimized in BPSO is evaluated as a linear combination of accuracy (A), and the ratio of the number of selected features ($N_{selected}$) to the total features in the dataset (N_{total}). Hence, the BPNN classifier with k -fold cross-validation is employed to measure the accuracy of the selected features. Improved accuracy and reduced features are considered as an objective function to optimize the evaluated fitness value using Eq. (9)

$$Fitness = f(x^m(n)) = \alpha A(n) + (1 - \alpha) \left(1 - \frac{N_{selected}(n)}{N_{total}} \right), \quad (9)$$

where α is the weighing factor that decides the trade-off between the accuracy and the selected features. In this work, α value is set to be 0.8. Therefore, it is clear that the ratio of accuracy and the selected features involvement in the fitness function evaluation is 8:2. After evaluating the fitness function, the particle's personal and global best positions are updated using Eqs. (10) and (11).

$$x_p^m(n) = \begin{cases} x^m(n) & \text{if } f(x^m(n)) > f(x_p^m(n)) \\ x_p^m(n) & \text{if } f(x^m(n)) \leq f(x_p^m(n)) \end{cases} \quad (10)$$

$$x_G(n) = \begin{cases} x_p^m(n) & \text{if } f(x_p^m(n)) > f(x_G(n)) \\ x_G(n) & \text{if } f(x_p^m(n)) \leq f(x_G(n)) \end{cases} \quad (11)$$

Predefined boundary limits are applied in both the particle's position and velocity.

2.5.2 Time-varying acceleration coefficients BPSO (TVAC-BPSO) based classifier

Acceleration coefficients namely cognitive parameter (c_1) and social parameter (c_2) are responsible for the particles flying towards the desirable regions. Improper selection of acceleration coefficients leads to the particles flying towards the undesirable regions. To avoid this, c_1 and c_2 values are set to be 2 in the standard PSO. Sometimes these values may be unsuccessful in the overall searching of the optimal solution [28]. The searching capability of the particles gets improved by introducing time-varying acceleration coefficients [29] in BPSO (TVAC-BPSO) to reach the optimal solution in search space. In the initial stage of an algorithm, the coefficient values ($c_1 > c_2$) have been used for the global search and then in the last stage, the coefficient values ($c_1 < c_2$) are set for the localized search. The time-varying form of acceleration coefficients $c_1(n)$ and $c_2(n)$ are represented in Eqs. (12) and (13)

$$c_1(n) = \left(\frac{(c_{1max} - c_{1min}) \times (Max_n - n)}{Max_n} \right) + c_{1min}, \quad (12)$$

$$c_2(n) = \left(\frac{(c_{2max} - c_{2min}) \times (Max_n - n)}{Max_n} \right), \quad (13)$$

where c_{1min} and c_{1max} are the minimum and maximum values of the cognitive parameters, c_{2min} and c_{2max} are the minimum and maximum values of the social parameters, Max_n and n are the maximum and the current iteration number.

2.5.3 Time-Varying Inertia Weight BPSO (TVIW-BPSO) based classifier

The performance of the BPSO is improved by selecting inertia weight with linearly decreasing values from 0.9 to 0.4 [30] using Eq. (14).

$$w(n) = \left(\frac{(w_{max} - w_{min}) \times (Max_n - n)}{Max_n} \right) + w_{min} \quad (14)$$

where w_{min} and w_{max} are the lowest and highest inertia weights, Max_n and n are the maximum and the current iteration number.

The non-linear inertia weight is obtained by using Eq. (15)

$$w(n) = \left(\frac{(w_{max} - w_{min}) \times (Max_n - n)^c}{(Max_n)^n} \right) + w_{min} \quad (15)$$

where c is an exponential factor that influences the nonlinearity [31] in the linearly decreasing inertia weight.

2.5.4 Evolutionary state greedy reset BPSO (EBPSO) based classifier

In BPSO, the optimal feature subsets are described by the particle's position in the search space. The resultant feature subset may not be optimal if particles get trapped in a suboptimal solution due to the premature convergence problem. Hybridization of PSO, adaptive inertia weight and acceleration coefficient update mechanisms [32-35] are introduced to avoid early convergence problems. But it fails to discriminate the features of defective and defect-free fabrics. To offset this, an Evolutionary state greedy reset BPSO (EBPSO) is introduced to obtain the optimal feature subset by evaluating the fitness function using back propagation neural network (BPNN).

In this work, an evolutionary state parameter $E^m(n)$ is introduced in BPSO to obtain unique control parameters. The control parameters such as inertia weight and the acceleration coefficients have a significant role in the effective performance of BPSO. In most of the research, a high range of inertia weight is used in the beginning and it gradually decreases in further iterations. However, this time-varying inertia weight may not be effective for global search and thus novel self-adaptive control parameters are proposed in the BPSO based classifier for an effective search of the algorithm. In this, each particle has unique control parameters. The uniqueness of the control parameters is achieved using a feedback parameter that is controlled by the evolutionary state. In an evolutionary state calculation, only the feasible solution of the particle is considered and this promotes better learning within the swarm. This is shown in Eq. (16)

$$E^m(n) = \frac{f(x_G(n)) - f(x_p^m(n))}{f(x_f^m(n))}, \quad (16)$$

where $f(x_P^m(n))$ and $f(x_G(n))$ are the particle's personal and global best fitness values, $f(x_f^m(n))$ is the most recent feasible solution of the particle. In this approach evolutionary state is used to accelerate the cognitive and social parameters to identify the optimal solutions based on their hyperspace location as given by Eqs. (17) and (18).

$$c_1^m(n) = \begin{cases} \left(\frac{(c_{\max} - c_{\min}) \times (Max_n - n)}{Max_n} \right) + c_{\min} & \text{if } 0 \leq E^m(n) \leq 0.5 \\ c_{\max} - \left(\frac{(c_{\max} - c_{\min}) \times (Max_n - n)}{Max_n} \right) & \text{if } 0.5 \leq E^m(n) \leq 1.0 \end{cases} \quad (17)$$

$$c_2^m(n) = \begin{cases} c_{\max} - \left(\frac{(c_{\max} - c_{\min}) \times (Max_n - n)}{Max_n} \right) & \text{if } 0 \leq E^m(n) \leq 0.5 \\ \left(\frac{(c_{\max} - c_{\min}) \times (Max_n - n)}{Max_n} \right) + c_{\min} & \text{if } 0.5 \leq E^m(n) \leq 1.0 \end{cases} \quad (18)$$

In this adaptive approach, global search is promoted by the low-level evolutionary state (between 0 and 0.5) in the initial iterations. In the later iterations, local search is promoted by exploring the regions around its best historical solutions. A high-level evolutionary state (between 0.5 and 1) promotes more exploitation in the search space during initial iterations and finally, it converges to the swarm best solution. As a result, particles adapt solely according to the feasible solution and thus it promotes the effective learning of the particles in the search space.

In the proposed approach, an evolutionary state is used instead of inertia weight for updating the velocity of the particles to reach the best solutions without any deviation. Particle's velocity is updated based on Eq. (19)

$$v^m(n+1) = E^m(n)v^m(n) + c_1(n)rand_1(n)(x_p^m(n) - x^m(n)) + c_2(n)rand_2(n)(x_G(n) - x^m(n)) - (1 - E^m(n))(x_G(n) - x_p^m(n)). \quad (19)$$

Repulsion of the particle is based on the evolutionary state as well as the difference between swarm best and particle's personal best solutions. Higher repulsion is achieved with lower evolutionary states and lower repulsion is achieved with the higher evolutionary state.

Also, swarm best re-initialization is introduced to obtain the optimal feature subset if the fitness value is constant for a fixed number of runs. The swarm best positions are re-initialized by employing mutation and crossover operation to improve the swarm best solution. Therefore, quality of the whole population will be improved by attracting the particle's personal best solutions towards the swarm best solution. The mutation operation is performed to obtain trivial position from the swarm best position using Eq. (20).

$$y^m(n) = x_G(n) + (b \times (x_{r1}(n) - x_{r2}(n))) \quad (20)$$

where $x_{r1}(n)$ and $x_{r2}(n)$ are the positions of the randomly chosen neighboring particles and b is the scaling factor and it is set be 0.2.

The crossover operation is performed between the trivial position $y^m(n)$ and the

current best position of the particle $x^m(n)$ to obtain re-initialized position using Eq. (21).

$$x^m(n+1) = \begin{cases} y^m(n) & \text{if } rand \leq pCR \\ x^m(n) & \text{if } rand > pCR \end{cases} \quad (21)$$

where PCR is a crossover constant and it is set to be 0.2.

Pseudocode of the proposed evolutionary state greedy reset BPSO based classifier

```

begin
algorithm parameters
for each particle in the swarm do
    find the evolutionary state using Eq. (16)
    update acceleration coefficients using Eqs. (17) and (18)
    update particle's velocity using Eq. (19)
    evaluate fitness function using BPNN
    revise personal and global best positions based on the evaluated fitness
    if fitness value is constant for a fixed number of runs
        perform mutation on global best position using Eq. (20)
        re-initialize the position by performing crossover between  $y^m(n)$  and
         $x^m(n)$  using Eq. (21)
    end
end
write global best solution of the selected features;
```

3. IMPLEMENTATION AND EXPERIMENTAL RESULTS

The performance of the proposed method is analyzed with TILDA textile image database. TILDA database has eight different textured fabrics and everything categorized into four directories namely C1, C2, C3, and C4. Each directory is divided into two sub-class directories namely R1 and R3. These subdirectories (R1 and R3) are further divided into eight more subdirectories namely E1, E2, E3, E4, E5, E6, and E7, and each of which contains 50 fabric images. In this defect-free fabric images are in the subdirectory E0, whereas defective fabric images are in the other subdirectories (E1-E7). Therefore, a total of 400 fabric images are considered from each class directories for the experimental analysis. In this work, an experiment was conducted for the fabric with a complex pattern (non-p1 group) which was taken from the class directories C3R1, C3R3, C4R1, and C4R3. Data set details are listed in Table 2. All the datasets listed in Table 2 represent a binary-class problem with 22 features.

For the analysis of this work, different combinations of feature extraction techniques explained in section 2 have been used to check their variability among the defective and defect-free fabrics. The co-occurrence features were extracted from (i) GLCM (ii) combination of the Wavelet transform and GLCM (W-GLCM) (iii) combination of the Wavelet transform, local binary pattern, and GLCM (WL-GLCM). These extracted features are normalized using the Min-Max technique to scale the data between 0 and 1. Table 3 shows the sample co-occurrence features of defective (D) and defect-free (ND) fabrics (Average (M) and standard deviation (SD) values of the corresponding features in the group) extracted from the C3R1 group of data set.

Table 2. Database description.

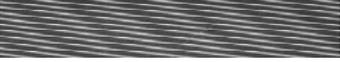
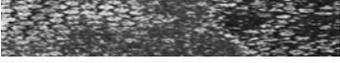
D#	Data Set	Fabric Sample	#Features	# Instances
D1	C3R1		22	400
D2	C3R3		22	400
D3	C4R1		22	400
D4	C4R3		22	400

Table 3. Sample co-occurrence features for C3R1 group of fabrics.

Co-occurrence features	WL-GLCM		W-GLCM		GLCM	
	D	ND	D	ND	D	ND
	M±SD	M±SD	M±SD	M±SD	M±SD	M±SD
Contrast	0.44±0.25	0.94±0.03	0.40±0.22	0.32±0.24	0.59±0.16	0.62±0.13
Correlation(M)	0.82±0.16	0.22±0.12	0.40±0.20	0.32±0.24	0.49±0.16	0.53±0.13
Correlation	0.82±0.16	0.22±0.12	0.40±0.20	0.32±0.24	0.49±0.16	0.53±0.13
Cluster prominence	0.79±0.18	0.80±0.01	0.41±0.15	0.34±0.15	0.44±0.12	0.48±0.06
Cluster Shade	0.57±0.30	0.75±0.02	0.55±0.17	0.46±0.20	0.45±0.14	0.45±0.09
Dissimilarity	0.44±0.25	0.94±0.03	0.41±0.21	0.32±0.23	0.68±0.16	0.72±0.12
Energy	0.27±0.29	0.13±0.04	0.45±0.21	0.57±0.23	0.19±0.11	0.15±0.06
Entropy	0.74±0.27	0.93±0.02	0.55±0.19	0.42±0.19	0.76±0.11	0.80±0.05
Homogeneity(M)	0.55±0.25	0.05±0.03	0.56±0.20	0.66±0.22	0.26±0.16	0.22±0.12
Homogeneity	0.55±0.25	0.05±0.03	0.56±0.20	0.66±0.22	0.28±0.16	0.25±0.12
Max Probability	0.41±0.29	0.38±0.04	0.56±0.20	0.66±0.21	0.32±0.14	0.26±0.10
Sum of Squares: Variance	0.43±0.25	0.93±0.03	0.59±0.20	0.68±0.22	0.50±0.17	0.65±0.07
Sum average	0.43±0.25	0.93±0.03	0.60±0.20	0.69±0.22	0.57±0.16	0.71±0.06
Sum variance	0.43±0.25	0.93±0.03	0.60±0.20	0.69±0.22	0.57±0.16	0.71±0.06
Sum Entropy	0.75±0.27	0.90±0.03	0.57±0.19	0.44±0.18	0.73±0.11	0.76±0.04
Difference Variance	0.44±0.25	0.94±0.03	0.40±0.22	0.32±0.24	0.59±0.16	0.62±0.13
Difference Entropy	0.53±0.25	0.96±0.02	0.57±0.19	0.45±0.19	0.70±0.14	0.73±0.09
Information Correlation 1	0.82±0.13	0.46±0.05	0.39±0.21	0.35±0.28	0.30±0.18	0.32±0.12
Information Correlation 2	0.89±0.14	0.76±0.05	0.49±0.21	0.40±0.25	0.47±0.16	0.53±0.10
Inverse Difference normalized	0.55±0.25	0.05±0.03	0.57±0.21	0.66±0.23	0.30±0.16	0.26±0.12
Inverse Difference Moment normalized	0.55±0.25	0.05±0.03	0.59±0.21	0.67±0.24	0.40±0.16	0.36±0.13
Autocorrelation	0.43±0.25	0.93±0.03	0.59±0.21	0.68±0.23	0.50±0.17	0.65±0.07

For the analysis of the proposed work, 400 fabric images from each group (350 defective fabric, and 50 non-defective fabric images) were used. In this work, BPNN with 10 hidden layer neurons is used. Gradient descent momentum and adaptive learning rate is used as an activation function to update weight and bias in the network. In swarm intelligence-based classifiers, the fitness function is evaluated by BPNN on the selected feature subset using 10-fold cross-validation with 90% of the images for training and 10% of the images for testing. This process will be repeated to test all the input images. In this work, the maximum number of runs to implement an algorithm is set to be 10, acceleration coefficients c_{\max} and c_{\min} values are set to be 4 and 0 respectively. To evaluate the selected features, the target value has to be set for both defective and non-defective fabrics.

For defective fabric, lower magnitude scale of 0-1 is selected as a target T_D based on Eq. (22)

$$\frac{1}{N} \sum_{i=1}^L m_i \geq T_D \tag{22}$$

where m_i represent the averaged features (normalized) of L defective fabric images used for classification.

For non-defective fabric, higher magnitude scale of 0-1 is selected as a target T_{ND} based on Eq. (23)

$$\frac{1}{N} \sum_{j=1}^M m_j \geq T_{ND} \tag{23}$$

Where m_j represent the averaged features (normalized) of M defect-free fabric images used for classification. The target values (T_D and T_{ND}) are set such that the magnitude difference between them should be greater than or equal to 0.5 ($|T_{ND} - T_D| \geq 0.5$).

By considering the above constraints, the target values selected for the defective (T_D) and defect-free fabrics (T_{ND}) are 0.05 and 0.95 respectively.

There are many popular techniques are available to evaluate the performance of different classifiers. A Confusion matrix is one of the commonly used methodologies. This is shown in Table 4. True positive (TP) indicates that the defective fabric is detected as defective one, whereas true negative (TN) denotes the number of defect-free fabric is detected as defect-free fabric, therefore correct classification results are stored in the true positive and true negative. False-positive (FP) is a result in which the number of defect-free fabric is detected as defective fabric, whereas false negative (FN) is a result in which defective fabric is detected as defect-free fabric, thus in FP and FN incorrect classification results are stored.

Table 4. Confusion Matrix of defect detection.

		Actual	
		D	ND
Predicted	D	TP	FP
	ND	FN	TN

Table 5 shows the resultant confusion matrix obtained by the BPSO, TVAC-BPSO, TVIW-BPSO, and EBPSO based classifiers for the GLCM features, W-GLCM features, and WL-GLCM features for the different group of texture in the TILDA database.

Table 5. Confusion matrix obtained by the swarm intelligence based classifier.

Method	D#	Confusion Matrix											
		GLCM based co-occurrence features				W-GLCM based co-occurrence features				WL-GLCM based co-occurrence features			
		<i>TP</i>	<i>TN</i>	<i>FP</i>	<i>FN</i>	<i>TP</i>	<i>TN</i>	<i>FP</i>	<i>FN</i>	<i>TP</i>	<i>TN</i>	<i>FP</i>	<i>FN</i>
BPSO based Classifier	D1	338	25	25	12	347	3	48	2	347	50	1	2
	D2	350	1	49	0	350	1	49	0	347	50	1	2
	D3	350	2	48	0	349	1	50	0	348	50	0	2
	D4	339	17	34	10	347	1	50	2	347	50	1	2
TVAC-BPSO based classifier	D1	333	24	27	16	343	5	46	4	349	49	1	1
	D2	346	0	50	4	349	0	50	1	349	49	1	1
	D3	341	3	47	9	350	0	50	0	347	50	0	3
	D4	337	4	47	12	344	5	46	5	348	50	1	1
TVIW-BPSO based classifier	D1	331	27	24	18	346	4	47	3	349	49	1	1
	D2	346	0	50	4	348	0	50	2	349	49	1	1
	D3	350	1	49	0	350	0	49	1	349	49	1	1
	D4	331	27	24	18	347	1	50	2	349	50	1	0
EBPSO based classifier	D1	339	21	30	10	348	5	46	1	349	50	0	1
	D2	348	0	50	2	349	0	50	1	349	50	0	1
	D3	347	4	46	3	349	1	49	1	349	50	0	1
	D4	336	9	42	13	347	1	50	2	349	50	1	0

From Table 5, it shows that the classifier results are mapped towards the positive class in the resultant confusion matrix for the GLCM and W-GLCM based co-occurrence features. This is because, the more positive class images are present in the dataset than in the negative class images, and thus results are more biased towards the positive class. Above said co-occurrence features fails to discriminate positive as well as the negative class data sets. To discriminate both the classes, LBP is obtained before extracting co-occurrence features by exploring local textural information of the dataset in each class. From Table 5, it is clear that after including LBP in the feature extraction process (WL-GLCM) true negative rate is increased by reducing the false positive rate.

From the confusion matrix, the following performance evaluation metrics are calculated and analyzed for their effectiveness in the classifier performance.

Accuracy (Acc) is a measure of the defect detection rate of the classifier. This is shown in Eq. (24)

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%. \quad (24)$$

Misclassified result of the classifier is measured in terms of error rate (ER); it is described in Eq. (25)

$$ER = \frac{FN + FP}{TP + TN + FP + FN} \times 100\%. \quad (25)$$

Sensitivity (Se) of the classifier measures how well the classifier will classify the defective fabric when it is defective. The best and worst-case sensitivity measures are 100% and 0% respectively.

$$Se = \frac{TP}{TP + FN} \times 100\% \quad (26)$$

Specificity (Sp) of the classifier measures how well the classifier will classify the defect-free fabric when it is defect-free fabric. The best and worst case specificity measures are 100% and 0% respectively.

$$Sp = \frac{TN}{TN + FP} \times 100\% \quad (27)$$

The receiver operating characteristic curve (ROC) illustrates how good the model will be able to separate two classes, and thus often used for binary classification problems. The area under ROC (AUC) is a performance metric that describes how well the positive class probabilities are separated from the negative class probabilities [36]. Hence AUC in Eq. (28) is the average measure of correctly predicted results (TPR and TNR).

$$AUC = \frac{1}{2} \left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP} \right) \quad (28)$$

Precision (Pr) or positive predictive value (PPV) is a measure of how well the system is at correctly predicting defective fabric among all the predicted results as a defective fabric. That is, how accurate the defect detection in the fabric is

$$Pr = \frac{TP}{FP + TP} \times 100. \quad (29)$$

Negative predictive value (NPV) is a measure of how well the system is at correctly predicting defect-free fabric among the predicted result as defect-free fabric.

$$NPV = \frac{TN}{FN + TN} \times 100 \quad (30)$$

False-positive rate (FPR) is a measure of how the model is at predicting positive events when it is negative events. The best and worst cases FPR are 0 and 1 respectively.

$$FPR = \frac{FP}{FP + TN} \times 100 = 1 - Specificity \quad (31)$$

True negative rate (TNR) is a measure of how well the model is at predicting negative results correctly (defect-free fabric) among all the negative events.

$$TNR = \frac{TN}{FP + TN} = 1 - FPR \quad (32)$$

F -score is the weighted average score of precision and recall. This is used to get the best precision and recall at the same time and it ranges from 0 to 1. It tells about the classifier that how precisely it classifies the instances correctly.

$$F_{\beta} = \frac{(1 + \beta^2)(Precision \times Recall)}{(\beta^2 Precision + Recall)} \quad (33)$$

Commonly used β values are 0.5, 1 and 2. In this work, the F_1 score ($\beta = 1$) is measured to analyze the performance of the classifier.

Table 6 shows the performance evaluation metrics obtained from the BPSO based BPNN classifier.

BPSO based BPNN classifier is evaluated for the three different co-occurrence feature combinations. Among these three feature combinations; WL-GLCM based co-occurrence features yield better classification results than other features. Even though the accuracy percentage of the classifier for GLCM and W-GLCM based co-occurrence feature is more than 80%, but their false positive rate is more than 50%. This is because the dataset used to analyse this method is unbalanced (350 defective fabric and 50 non-defective fabric) and also the features extracted from these two methods will not contain local texture information of the input images. Hence most of the fabric samples are predicted as a defective fabric (positive class). From the above-mentioned feature extraction techniques, LBP contains well described local texture information. So that features extracted from the WL-GLCM tries to reduce false-positive rates. This is shown in Table 6.

Tables 7 and 8 show the performance evaluation metrics obtained from the TVAC-BPSO and TVIW-BPSO based BPNN classifier.

Table 6. Performance evaluation metrics of BPSO based BPNN.

D#	Features	Classifier Performance Metrics									
		Acc (%)	ER (%)	Se (%)	Sp (%)	AUC (%)	Pr (%)	NPV (%)	FPR (%)	TNR (%)	F ₁ Score (%)
D1	GLCM	90.75	9.25	96.6	50	73.3	93.1	67.6	50	50	94.8
	W-GLCM	87.5	12.5	99.4	5.82	52.7	87.9	60	94.1	5.9	93.3
	WL-GLCM	99.25	0.75	99.4	98.0	98.7	99.7	96.2	2	98.0	99.6
D2	GLCM	87.75	12.25	100	2	51	87.7	100	98	2	93.5
	W-GLCM	87.75	12.25	100	2	51	87.7	100	98	2	93.5
	WL-GLCM	99.25	0.75	99.4	98.0	98.7	99.7	96.2	1.96	98.0	99.6
D3	GLCM	88.0	12	100	4	52	87.9	100	96	4	93.6
	W-GLCM	87.5	12.5	100	1.96	51.0	87.5	100	98.0	1.96	93.3
	WL-GLCM	99.5	0.5	99.4	100	99.7	100	96.2	0	100	99.7
D4	GLCM	89.0	11	97.1	33.3	65.2	90.9	63.0	66.7	33.3	93.9
	W-GLCM	87.0	13	99.4	1.96	50.7	87.4	33.3	98.0	1.96	93.0
	WL-GLCM	99.25	0.75	99.4	98.0	98.7	99.7	96.1	1.96	98.0	99.6

Table 7. Performance evaluation metrics of TVAC-BPSO based BPNN.

D#	Features	Classifier Performance Metrics									
		Acc (%)	ER (%)	Se (%)	Sp (%)	AUC (%)	Pr (%)	NPV (%)	FPR (%)	TNR (%)	F ₁ Score (%)
D1	GLCM	89.25	10.75	95.4	47.1	71.2	92.5	60	52.9	47.1	93.9
	W-GLCM	87.5	12.5	98.9	9.8	54.3	88.2	55.6	90.2	9.8	93.2
	WL-GLCM	99.5	0.5	99.7	98	98.9	99.7	98	2	98	99.7
D2	GLCM	86.5	13.5	98.9	0	49.4	87.4	0	100	0	92.6
	W-GLCM	87.25	12.75	99.7	0	49.9	87.5	0	100	0	93.2
	WL-GLCM	99.5	0.5	99.7	98	98.9	99.7	98	2	98	99.7
D3	GLCM	86	14	97.4	6	51.7	87.9	25	94	6	92.4
	W-GLCM	87.5	12.5	100	2.0	51.0	87.5	100	98.0	2.0	93.3
	WL-GLCM	99.25	0.75	99.1	100	99.6	100	94.3	0	100	99.6
D4	GLCM	85.25	14.75	96.6	7.84	52.2	87.8	25	92.2	7.8	92.0
	W-GLCM	87.25	12.75	98.6	9.80	54.2	88.2	50	90.2	9.8	93.1
	WL-GLCM	99.5	0.5	99.7	98.0	98.9	99.7	98.0	2.0	98.0	99.7

Table 8. Performance evaluation metrics of TVIW-BPSO based BPNN.

D#	Features	Classifier Performance Metrics									
		Acc (%)	ER (%)	Se (%)	Sp (%)	AUC (%)	Pr (%)	NPV (%)	FPR (%)	TNR (%)	F ₁ Score (%)
D1	GLCM	89.5	10.5	94.8	52.9	73.9	93.2	60	47.1	52.9	94.0
	W-GLCM	87.5	12.5	99.1	7.8	53.5	88.0	57.1	92.2	7.8	93.3
	WL-GLCM	99.5	0.5	99.7	98	98.9	99.7	98	2	98	99.7
D2	GLCM	86.5	13.5	98.9	0	49.4	87.4	0	100	0	92.8
	W-GLCM	87	13	99.4	0	49.7	87.4	0	100	0	93.0
	WL-GLCM	99.5	0.5	99.7	98	98.9	99.7	98	2	98	99.7
D3	GLCM	87.75	12.25	100	2	51	87.7	100	98	2	93.5
	W-GLCM	87.5	12.5	99.7	0	49.9	87.7	0	100	0	93.3
	WL-GLCM	99.5	0.5	99.7	98	98.9	99.7	98	2	98	99.7
D4	GLCM	85.5	14.5	96.9	7.84	52.4	87.8	26.6	92.1	7.84	92.1
	W-GLCM	87	13	99.4	1.96	50.7	87.4	33.3	98.0	1.96	93.0
	WL-GLCM	99.75	0.25	100	98.0	99.0	99.7	100	1.96	98.0	99.9

The results obtained from TVAC-BPSO based BPNN and TVIW-BPSO based BPNN are compared with the standard BPSO. From the results, it can be concluded that there is no significant improvement in classifier performance. Hence EBPSO based classifier is proposed to get better performance results with reduced feature subset. In this method, social and cognitive parameters are updated based on the evolutionary state. These time-varying coefficients stimulate the particles in the swarm to escape from the local convergence problem and thus an improvement in the performance of the proposed method is observed. This is shown in Table 9.

Table 9. Performance evaluation metrics of EBPSO based BPNN.

D#	Features	Classifier Performance Metrics									
		Acc (%)	ER (%)	Se (%)	Sp (%)	AUC (%)	Pr (%)	NPV (%)	FPR (%)	TNR (%)	F ₁ Score (%)
D1	GLCM	90	10	97.1	41.9	69.2	91.9	67.7	58.8	41.2	94.4
	W-GLCM	88.25	11.75	99.7	9.80	54.8	88.3	83.3	90.2	9.80	93.7
	WL-GLCM	99.75	0.25	99.7	100	99.9	100	98.0	0	100	99.9
D2	GLCM	87	13	99.4	0	49.7	87.4	0	100	0	93.1
	W-GLCM	87.25	12.75	99.7	0	49.9	87.5	0	100	0	93.2
	WL-GLCM	99.75	0.25	99.7	100	99.9	100	98.0	0	100	99.9
D3	GLCM	87.75	12.25	99.1	8	53.6	88.3	57.1	92	8	93.4
	W-GLCM	87.5	12.5	99.7	2	50.9	87.7	50	98	2	93.3
	WL-GLCM	99.75	0.25	99.7	100	99.9	100	98.0	0	100	99.9
D4	GLCM	86.25	13.75	96.3	17.7	57.0	88.9	40.9	82.3	17.7	92.4
	W-GLCM	87	13	99.4	2.0	50.7	87.4	33.3	98.0	1.96	93.0
	WL-GLCM	99.75	0.25	100	98.0	99.0	99.7	100	2.0	98.0	99.9

By comparing all the performance metrics in Tables 6-9, WL-GLCM based co-occurrence features yield better results when compared to GLCM and W-GLCM features. Hence WL-GLCM based co-occurrence features alone considered for the comparison of proposed and existing swarm intelligence-based classifiers. This is shown in Table 10.

Table 10. Performance evaluation of the Swarm Intelligence based BPNN (SINN) for WL-GLCM co-occurrence features.

D#	Features	Classifier Performance Metrics									
		Acc (%)	ER (%)	Se (%)	Sp (%)	AUC (%)	Pr (%)	NPV (%)	FPR (%)	TNR (%)	F1 Score (%)
D1	BPSO	99.25	0.75	99.4	98.0	98.7	99.7	96.2	2.0	98.0	99.6
	TVAC-BPSO	99.5	0.5	99.7	98	98.9	99.7	98	2	98	99.7
	TVIW-BPSO	99.5	0.5	99.7	98	98.9	99.7	98	2	98	99.7
	EBPSO	99.75	0.25	99.7	100	99.9	100	98.0	0	100	99.9
D2	BPSO	99.25	0.75	99.4	98.0	98.7	99.7	96.2	2.0	98.0	99.6
	TVAC-BPSO	99.5	0.5	99.7	98	98.9	99.7	98	2	98	99.7
	TVIW-BPSO	99.5	0.5	99.7	98	98.9	99.7	98	2	98	99.7
	EBPSO	99.75	0.25	99.7	100	99.9	100	98.0	0	100	99.9
D3	BPSO	99.5	0.5	99.4	100	99.7	100	96.2	0	100	99.7
	TVAC-BPSO	99.25	0.75	99.1	100	99.6	100	94.3	0	100	99.6
	TVIW-BPSO	99.5	0.5	99.7	98	98.9	99.7	98	2	98	99.7
	EBPSO	99.75	0.25	99.7	100	99.9	100	98.0	0	100	99.9
D4	BPSO	99.25	0.75	99.4	98.0	98.7	99.7	96.2	2.0	98.0	99.6
	TVAC-BPSO	99.5	0.5	99.7	98.0	98.9	99.7	98.0	2.0	98.0	99.7
	TVIW-BPSO	99.5	0.5	99.7	98	98.9	99.7	98	2	98	99.7
	EBPSO	99.75	0.25	100	98.0	99.0	99.7	100	2.0	98.0	99.9

By comparing the performance of the swarm intelligence-based classifier, the proposed EBPSO based classifier yields better results than the existing methods without compromising any other performance metric.

Accuracy and error rates are good measures for balanced data classes. But for an unbalanced data set, accuracy alone is not a good measure. Therefore, the Matthews Correlation Coefficient [37, 38] and kappa values are calculated and analysed for an unbalanced dataset. *MCC* is calculated based on the confusion matrix entries using Eq. (34).

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (34)$$

In *MCC*, a high score is generated if the binary classifier predicts both positive and negative data instances correctly. The coefficient range will be in the interval $[-1, +1]$, and it will reach $+1$ for the best case, whereas it will reach -1 for the worst-case problem.

Kappa coefficient: The Cohen's kappa coefficient (k) is another metric used to measure the agreement between two classes and it is useful for all instances of data; the two classes in this study are defective fabric class and defect-free fabric class. Besides *kappa* is less uncertain than that of using the accuracy metric, because it uses random chance and it is measured on a scale of 0-1. When k is closer to 0 indicates the poor agreement among the classifiers whereas closer to 1 indicates good agreement among the classifier [39].

The kappa coefficient (k) is calculated from the observed agreement (p_o) and the hypothetical probability of chance agreement (p_e) using Eqs. (35) and (36)

$$p_o = \frac{TP + TN}{T}, \quad (35)$$

$$p_e = \left(\frac{TP + FN}{T} \times \frac{TP + FP}{T} \right) + \left(\frac{TN + FN}{T} \times \frac{TN + FP}{T} \right), \tag{36}$$

where $T = TP + TN + FP + FN$. Thus, binary classifier is validated with kappa coefficient using Eq. (37)

$$k = \frac{p_o - p_e}{1 - p_e}. \tag{37}$$

Table 11. Performance analysis of proposed vs. literature on BPSO based classifiers for WL-GLCM features.

Dataset D#	SINN Algorithm	Classifier Performance Metrics			Ref
		MCC	Kappa, <i>k</i>	Average Features	
D1	BPSO	0.96663	0.96657	15	[27]
	TVAC-BPSO	0.977	0.977	22	[29]
	TVIW-BPSO	0.977	0.977	13	[40]
	EBPSO	0.989	0.989	5	-
D2	BPSO	0.96663	0.96657	22	[27]
	TVAC-BPSO	0.977143	0.977143	11	[29]
	TVIW-BPSO	0.977143	0.977143	22	[40]
	EBPSO	0.988732	0.988669	6	-
D3	BPSO	0.978	0.978	12	[27]
	TVAC-BPSO	0.967	0.967	12	[29]
	TVIW-BPSO	0.977	0.977	8	[40]
	EBPSO	0.989	0.989	6	-
D4	BPSO	0.967	0.967	22	[27]
	TVAC-BPSO	0.977527	0.977527	22	[29]
	TVIW-BPSO	0.977143	0.988669	22	[40]
	EBPSO	0.989	0.989	5	-

Table 11 shows the performance analysis of *MCC* and *Kappa* values of swarm intelligence-based classifiers for WL-GLCM based co-occurrence features.

The performance metrics *MCC* and *Kappa* coefficient values of the swarm intelligence-based classifiers are compared in Table 11. Among all the performance metrics, the most important performance metrics for the unbalanced dataset are *MCC* and *kappa*. By analysing the results, *MCC* and *kappa* values of existing methods are lower than that of the proposed method. Hence the proposed EBPSO based classifier gives better *MCC* and *kappa* values with the reduced number of features for all groups of fabrics in the given dataset. The results show that EBPSO based classifier overcomes the premature convergence problem of BPSO with improved results.

Statistical Analysis: Analysis of variance (ANOVA) [43] is one of the statistical analysis methods that find the significant differences among the mean values of more than two experimental results; whereas the *t*-test is limited to analyze the mean value of two experimental results. So in this work, one way ANOVA test is conducted for the statistical analysis of the swarm intelligence based classifiers. In an ANOVA test, a significant difference is assessed by finding the ratio of between-group variance (*MSB*) to the within-group variance (*MSW*). This ratio is known as *F (observed)* with $k - 1, N - k$ degrees of freedom (*df*).

This is shown in Eq. (38).

$$F(\text{observed}) = \frac{MSB}{MSW} = \frac{SSB/k-1}{SSW/N-k} \quad (38)$$

The statistical test result is obtained by comparing $F(\text{observed})$ with $F(\text{critical})$ at a significance level of $\alpha = 0.05$. If $F(\text{observed})$ value is greater than the $F(\text{critical})$ value, then there will be a significant difference among the experimental results. Table 12 shows the ANOVA test results conducted for the MCC value of the Swarm intelligence-based classifiers.

Table 12. ANOVA statistical test result for MCC value of the swarm intelligence-based classifier.

Source of Variation	Sum of Square	df	Mean Square	$F(\text{observed})$	p -value	$F(\text{critical})$
Between Groups	0.000807 (SSB)	3 ($k-1$)	0.000269 (MSB)	18.60908 (MSB/MSW)	8.28E-05	3.490295
Within Groups	0.000174 (SSW)	12 ($N-k$)	1.45E-05 (MSW)			
Total	0.000981	15				

From Table 12, it is clear that the $F(\text{observed})$ value is greater than that of $F(\text{critical})$ value. This comparison result supports that at least one classifier performance differs from all other classifier's performance. But, it fails to identify the better classifier for fabric defect detection. Hence the Least Significant Difference (LSD) test is conducted to determine the better classifier. The Least Significant Difference is calculated using Eq. (39).

$$LSD = \frac{\sqrt{2 \times MSW \times F_{1,N-k}}}{r} \quad (39)$$

calculated LSD value for $r = 4$ and $F_{1,N-k} = F_{1,12} = 4.75$ using Eq. (39) is 0.007385.

In the LSD test, if the absolute difference between the mean values of any two experimental results is greater than 0.007385, it may be concluded that the experimental results obtained from the classifiers are statistically different.

Table 13. Least Significant Difference test result for MCC value of the swarm intelligence-based classifier.

SINN Classifier	Dataset				Mean (\bar{X})	Absolute Difference, $ \bar{X}_i - \bar{X}_j $
	C3R1	C3R3	C4R1	C4R3		
BPSO	0.96657	0.96657	0.978	0.967	0.970523 (\bar{X}_1)	$ \bar{X}_1 - \bar{X}_2 = 0.00513$
TVAC-BPSO	0.977	0.977143	0.967	0.977527	0.97389 (\bar{X}_2)	$ \bar{X}_1 - \bar{X}_3 = 0.01042$
TVIW-BPSO	0.977	0.977143	0.977	0.988669	0.980937 (\bar{X}_3)	$ \bar{X}_1 - \bar{X}_4 = 0.01938$ $ \bar{X}_2 - \bar{X}_3 = 0.00529$
EBPSO	0.989	0.988669	0.989	0.989	0.98889 (\bar{X}_4)	$ \bar{X}_2 - \bar{X}_4 = 0.01425$ $ \bar{X}_3 - \bar{X}_4 = 0.00896$

The absolute differences between the mean values of MCC obtained from the swarm intelligence-based classifiers are shown in Table 13. From the Table 13, it is clear that the absolute differences between the mean values of MCC obtained from the EBPSO based BPNN classifier (\bar{X}_4) and other swarm intelligence based BPNN classifiers ($(\bar{X}_1), (\bar{X}_2), (\bar{X}_3)$) are greater than the observed LSD. By this comparison, it is concluded that the EBPSO based BPNN classifier performance result is statistically different from all other existing BPSO based classifiers. Therefore, EBPSO based BPNN classifier achieves better results than the existing classifiers for defect detection in fabrics.

4. CONCLUSION

The proposed system is investigated on a different combination of co-occurrence features and performance metrics of each feature is evaluated with swarm intelligence-based classifiers for fabric defect detection. Among all the co-occurrence features better performance results are obtained by WL-GLCM based co-occurrence features. By comparing the results of all the swarm intelligence-based classifiers, the proposed Evolutionary state greedy reset BPSO (EBPSO) based classifier yields better results. In addition to that, it overcomes the premature convergence problem in BPSO. Hence proposed technique shows significant improvement in performance metrics with a reduced number of features compared to other BPSO based classifiers. The effectiveness of the classifier performance for an unbalanced dataset is analyzed with the Matthews Correlation Coefficient (MCC) and the proposed classifier yields maximum MCC and accuracy of 0.989 and 99.75% with 5 features than existing methods.

REFERENCES

1. S. Guan, "Fabric defect detection using an integrated model of bottom up and top-down visual attention," *Journal of Textile Institute*, Vol. 107, 2015, pp. 215-224.
2. M. T. Habib, R. H. Faisal, M. Rokonzaman, and F. Ahmed, "Automated fabric defect inspection: A survey of classifiers," *International Journal in Foundation of Computer Science and Technology*, Vol. 4, 2014, pp. 17-25.
3. Y. Han and P. Shi, "An adaptive level-selecting wavelet transform for texture defect detection," *Image and Vision Computing*, Vol. 25, 2007, pp. 1239-1248.
4. C. Wang, J. Li, M. Chen, Z. He, and B. Zuo, "The obtainment and recognition of raw silk defects based on machine vision and image analysis," *Journal of the Textile Institute*, Vol. 107, 2016, pp. 316-326.
5. X. Yang, G. Pang, and N. Yung, "Discriminative training approaches to fabric defect classification based on wavelet transform," *Pattern Recognition*, Vol. 37, 2004, pp. 889-899.
6. J. Zhou and J. Wang, "A real-time computer vision-based platform for fabric inspection part 1: Algorithm," *Journal of the Textile Institute*, Vol. 106, 2016, pp. 1282-1292.
7. J.-F. Jing, S. Chen, and P.-F. Li, "Fabric defect detection based on golden image subtraction," *Coloration Technology*, Vol. 133, 2017, pp. 26-39.
8. N. T. Deotale and T. Sarode, "Texture fault detection technique: Comparative study and critical review," *International Journal of Computer and Mathematical Science*, Vol. 7, 2018, pp. 299-305.

9. J. H. Conway, H. Burgiel, and C. Goodman-Strauss, *The Symmetries of Things*, Boca Raton, CRC Press, FL, 2016.
10. L.-H. Ding, L. Xiao, Y.-W. Zhu, W. Liu, and Y. Liu, "Gabor filter based automatic textile defect detection," in *Proceedings of SPIE – The International Society for Optical Engineering*, Vol. 4875, 2007, pp. 789-795.
11. J. Jing, H. Zhang, and P. Li, "Improved Gabor filters for textile defect detection," *Advanced in Control Engineering and Information Science*, 2011, pp. 5010-5014.
12. X. Yang, G. Pang, and N. Yung, "Robust fabric defect detection and classification using multiple adaptive wavelets," *IEE Proceedings – Vision, Image, and Signal Processing*, Vol. 152, 2005, pp. 715-723.
13. G. Yu and S. V. Kamarthi, "A cluster-based wavelet feature extraction method and its application," *Engineering Applications of Artificial Intelligence*, Vol. 23, 2010, pp. 196-202.
14. R. A. Lizarraga-Morales, F. E. Correa-Tome, R. E. Sanchez-Yanez, and J. Cepeda-Negrete, "On the use of binary features in a rule-based approach for defect detection on patterned textiles," *IEEE Access*, Vol. 7, 2019, pp. 18042-18049.
15. J. Jing, J. Wang, P. Li, and Y. Li, "Automatic classification of woven fabric structure by using learning vector quantization," *Advanced in Control Engineering and Information Science*, Vol. 15, 2011, pp. 5005-5009.
16. J. Jing, M. Xu, and P. Li, "Automatic recognition of weave pattern and repeat for yarn-dyed fabric based on KFCM and IDMF," *Optik*, Vol. 126, 2015, pp. 2876-2883.
17. A. Kumar, "Computer-vision-based fabric defect detection: A Survey," *IEEE Transactions on Industrial Electronics*, Vol. 55, 2008, pp. 348-363.
18. J. L. Raheja, S. Kumar, and A. Chaudhary, "Fabric defect detection based on GLCM and Gabor filter: A comparison," *Optik*, Vol. 124, 2013, pp. 6469-6474.
19. S. G. Mallet, "A theory for multi resolution signal decomposition: The wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 11, 1989, pp. 674-693.
20. T. Ojala, M. Pietikainen, and T. Maenpaa, "Multi resolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, 2002, pp. 971-987.
21. R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-3, 1973, pp. 610-621.
22. Workgroup on texture analysis of DFG, *TILDA textile texture database*, <http://lmb.informatik.uni-freiburg.de/research/dfg-texture/tilda>.
23. Y. Zhang, Z. Lu, and J. Li, "Fabric defect detection and classification using Gabor filters and Gaussian mixture model," in *Proceedings of the 9th Asian Conference on Computer Vision*, LNCS Vol. 5995, 2010, pp. 635-644.
24. T. Barbu, "Gabor filter-based face recognition technique," in *Proceedings of the Romanian Academy, Series A*, Vol. 11, 2010, pp. 277-283.
25. T. Chang and J. Kuo, "Texture analysis and classification with tree-structured wavelet transform," *IEEE Transactions on Image Processing*, Vol. 2, 1993, pp. 429-441.
26. K. G. Krishnan and P. T. Vanathi, "An efficient texture classification algorithm using integrated discrete wavelet transform and local binary pattern features," *Cognitive Systems Research*, Vol. 52, 2018, pp. 267-274.

27. J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, 1997, pp. 4104-4108.
28. J. F. Schutte and A. A. Groenwold, "A study of Global optimization using particle swarms," *Journal of Global Optimization*, Vol. 31, 2005, pp. 93-108.
29. A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, Vol. 8, 2004, pp. 240-255.
30. R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proceedings of Congress on Evolutionary Computation*, 2000, pp. 84-88.
31. A. Chatterjee and P. Siarry, "Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization," *Computers & Operations Research*, Vol. 33, 2006, pp. 859-871.
32. C.-F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design," *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, Vol. 34, 2004, pp. 997-1006.
33. M. Taherkhani and R. Safabakhsh, "A novel stability-based adaptive inertia optimization weight for particle swarm," *Applied Soft Computing*, Vol. 38, 2016, pp. 281-295.
34. E. Mezura-Montes and C. A. Coello-Coello, "Constraint-handling in nature-inspired numerical optimization: Past, present and future," *Swarm and Evolutionary Computation*, Vol. 1, 2011, pp. 173-194.
35. H. Liu, Z. Cai, and Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization," *Applied Soft Computing*, Vol. 10, 2010, pp. 629-640.
36. A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, Vol. 30, 1997, pp. 1145-1159.
37. D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, Vol. 10, 2020, pp. 1-13.
38. S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using Matthews correlation coefficient metric," *PLoS One*, 2017, No. e0177678.
39. J. Cohen, "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, Vol. 1, 1960, pp. 37-46.
40. A. Nickabadi, M. M. Ebadzadeh, and R. Safabakhsh, "A novel particle swarm optimization algorithm with adaptive inertia weight," *Applied Soft Computing*, Vol. 11, 2011, pp. 3658-3670.
41. H. Y. T. Ngan, G. K. H. Pang, and N. H. C. Yung, "Performance evaluation for motif-based patterned texture defect detection," *IEEE Transactions on Automation Science and Engineering*, Vol. 7, 2010, pp. 58-72.
42. H. Y. T. Ngan and G. K. H. Pang, "Regularity analysis for patterned texture inspection," *IEEE Transactions on Automation Science and Engineering*, Vol. 6, 2009, pp. 131-144.
43. E. Ostertagova and O. Ostertag, "Methodology and application of one-way ANOVA," *American Journal of Mechanical Engineering*, Vol. 1, 2013, pp. 256-261.

44. G. M. Nasira and P. Banumathi, "Plain woven fabric defect detection based on image processing and artificial neural networks," *International Journal of Computer Trends and Technology*, Vol. 6, 2013, pp. 226-229.
45. V. Gnanaprakash, P. T. Vanathi, and G. Suresh, "Defect detection in fabrics using back propagation neural networks," *International Journal of Innovative Technology and Exploring Engineering*, Vol. 8, 2018, pp. 132-138.
46. P. Vyas and M. Kakhani, "Fabric defect inspection system using neural network," *International Journal of Multidisciplinary Research and Development*, Vol. 2, 2015, pp. 569-573.
47. J. He and Q. Jiang, "Research on the fabric defect detection method based on improved PSO and NN algorithm," *International Journal of Digital Content Technology and its Applications*, Vol. 6, 2012, pp. 177-184.
48. Y. Li, H. Luo, M. Yu, G. Jiang, and H. Cong, "Fabric defect detection algorithm using RDPSO-based optimal Gabor filter," *The Journal of the Textile Institute*, Vol. 110, 2019, pp. 487-495.
49. A. V. Nalawade, P. H. Narkhede, S. B. Hande, G. K. Sarade, and R. S. Pisal, "Fabric inspection system using artificial neural networks," *International Journal of Advanced Research in Electronics and Communication Engineering*, Vol. 5, 2016, pp. 153-156.
50. M. S. Biradar and S. P. Jadhav, "Fabric defect detection by using neural network technique," *IOSR Journal of Electronics and Communication Engineering*, Vol. 9, 2014, pp. 70-76.
51. T. E. Mursalin, F. Z. Eishita, and A. R. Islam, "Fabric defect inspection system using neural network and microcontroller," *Journal of Theoretical and Applied Information Technology*, Vol. 4, 2008, pp. 560-570.



V. Gnanaprakash received his B.E. degree in Electronics and Communication Engineering from Hindustan College of Engineering, Chennai, and M.E. degree in Communication Systems from PSG College of Technology, Coimbatore in the year 2008 and 2013 respectively. He is currently pursuing a Ph.D. degree in the Department of Information and Communication Engineering, Anna University, Chennai. His research interests include image processing, evolutionary computing, and machine learning.



Ponnusamy Thangapandian Vanathi received her B.E. degree in Electronics and Communication Engineering and M.E. degree from PSG College of Technology, Coimbatore. She received her Ph.D. in the area of Speech Signal Processing in the year 2002 from Bharathiar University, Coimbatore. Her research interests include soft computing, speech signal processing, wireless sensor networks, and VLSI design. She is currently working as a Professor in ECE Department, PSG College of Technology, Coimbatore.