Received: February 21, 2022. Revised: May 6, 2022.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Effective PCB Defect Detection Using Stacked Autoencoder with Bi-LSTM Network

Jithendra P.R. Nayak[1]* **B.D. Parameshachari[1]**

[1]*GSSS Institute of Engineering & Technology for Women, Mysuru, India*
* Corresponding author's Email: jithendraatman.gmitece@gmail.com

Abstract: In the Printed Circuit Boards (PCBs) manufacturing, the defect detection is an important task that helps in improving the quality of PCBs production. The conventional defect detection schemes include few drawbacks like high computational cost, noise susceptibility, and strongly depends on a carefully designed template. To highlight the above stated drawbacks, a new hybrid deep learning model is proposed in this manuscript. Initially, the input PCB defective images are collected from the PCB defect dataset and further, the feature extraction is performed by utilizing Binary Robust Invariant Scalable Keypoints (BRISK) and Speeded up Robust Features (SURF) for extracting the feature vectors from the PCB defective image. The extracted feature vectors are multi-dimensional that increase the computational complexity, so the stacked autoencoder technique is applied for reducing dimension of the extracted feature vectors. The stacked autoencoder technique effectively selects minimal sub-set of non-redundant and relevant feature vectors and it is used for representing the datasets from original feature space to a reduced and more informative feature space. Finally, the selected feature vectors are fed to the Bi-Long Short Memory Network (Bi-LSTM) for classifying the defect types like mouse bite, spurious copper, short, spur, missing hole, and open circuit. The extensive experimental outcome confirmed that the hybrid deep learning model obtained higher performance in the PCB defect detection with the classification accuracy of 99.99%, which is superior related to the comparative models.

Keywords: Dimensionality reduction, Feature extraction, Long short memory network, PCB defect detection, Stacked autoencoder.

1. Introduction

In the recent decades, the electronic products are extensively utilized in human daily life like computer, television, mobile phone, etc., where the high quality PCBs are required for better usage of the electronic products [1-3]. During the PCB manufacturing process, several defects occur at different stages like scratches, holes, crack, broken edges, missing components or misaligned, etc. [4, 5]. The manual visual inspection is accomplished to find these defects in the electronic products. However, the manual visual defect detection is slow, inconsistent and subjective, where it does not satisfy the present manufacturing requirements [6-8]. Therefore, the machine learning techniques are effective for an automated PCB defect detection in the manufacturing process [9]. The existing machine learning techniques generally works based on texture analysis [10], where a set of active feature values are extracted from the spectral or the spatial domain of the test PCB defective image [11]. In addition to this, a high level multiple dimensional classification techniques such as random forest, support vector machine, etc. are employed for identifying the defect samples [12]. Whereas, the success of the classification techniques completely depends on the human experts for extracting and selecting the representative feature values based on the structure variations and local gray level variations of a defect in the test PCB defective image [13]. The machine learning techniques need handcrafted feature values, which is insufficient to detect the defect samples, so a new hybrid deep learning model is introduced in this manuscript, where the major contributions are given below:

- After the collection of PCB defective images, the feature extraction is carried out utilizing SURF and BRISK feature descriptors for extracting the discriminative feature vectors.
- Then, the stacked autoencoder is developed for diminishing the dimensions of the extracted features, which improves the computational complexity and training time of the Bi-LSTM classifier.
- Lastly, the dimensionally reduced features are fed to the Bi-LSTM classifier for classifying the defect types like mouse bite, spurious copper, short, spur, missing hole and open circuit. The proposed hybrid deep learning model's effectiveness is validated by means of f1-score, accuracy, specificity, Mathews Correlation Coefficient (MCC), and sensitivity.

This manuscript is organized as follows: few recent research papers on the topic "PCB defect detection" are reviewed in Section 2. The mathematical and experimental analysis of the proposed hybrid deep learning model is given in Sections 3 and 4. The conclusion of the proposed model is represented in Section 5.

2. Related works

Hassanin [14] developed an automated inspection model on the PCB images for precise location and identification of the fault types. In this literature, the SURF feature descriptor and morphological operation were employed for detecting feature points on the PCB images. Then, the Euclidean distance measure was utilized for feature matching to localize and detect the missing components regard-less of the PCB type, rotation, and position. However, the higher dimensional data were not evenly distributed by the Euclidean distance measure that was a main concern in this study. Kim [15] implemented an automated PCB inspection model named skip-connected convolutional autoencoder. The developed model was trained for decoding the defect images from the non-defect images. Next, the decoded images were matched with the original input images for identifying the defect location. Further, an image augmentation technique was used for improving the training performance of the presented model. The simulation result revealed that the presented model obtained promising performance in the PCB defect detection. In the inspection systems, the data imbalance was a major concern in the skip-connected convolutional autoencoder.

Adibhatla [16] implemented a new inspection model named You Only Look Once (YOLO) for detecting the defects in the PCB images. In this literature, around 11,000 PCB images and the network with 2 fully connected layers and 24 convolutional layers were utilized to achieve better PCB defect detection. Hu and Wang, [17] integrated feature pyramid network and faster Region-based Convolutional Neural Network (RCNN) to detect surface defects in the PCB images. The presented YOLO and RCNN models were computationally expensive, because it needs high-end graphics processing unit systems for data training and testing. Gaidhane [18] used a similarity measure named as symmetric matrix to compare the defect images from the non-defect images for an effective defect detection in the PCB images. As stated earlier, the multi-dimensional data were not evenly distributed by the similarity measure, which was a main problem in the defect detection.

Yuk [19] combined SURF feature descriptor and weighted kernel density estimation map to extract and choose the discriminative feature vectors for better defect detection in the PCB images. Further, the selected feature vectors were fed to the random forest classification technique for classifying the defect types. However, the random forest classifier was a weak learner related to other machine learning methods. Ding [20] combined Tiny Defect Detection Network (TDD-Net) and K-means clustering method to strengthen the relationship of feature maps for tiny defect detection. The quantitative examination on the PCB defect dataset showed that the presented model obtained superior defect detection performance compared to existing models. However, the presented TDD-Net model was suitable only for small defects. To highlight the issues mentioned in the literature phase, a new hybrid deep learning model (stacked autoencoder with Bi-LSTM) is proposed in this manuscript.

3. Methodology

In the PCB defect detection application, the proposed deep learning model includes four major steps like image collection: PCB defect dataset, feature extraction: BRISK and SURF descriptors, dimensionality reduction: stacked autoencoder, and

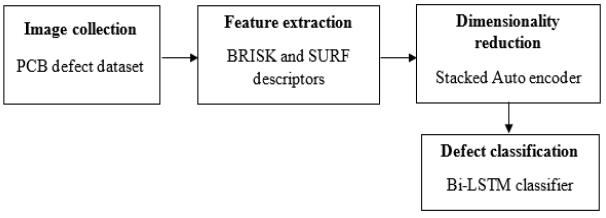
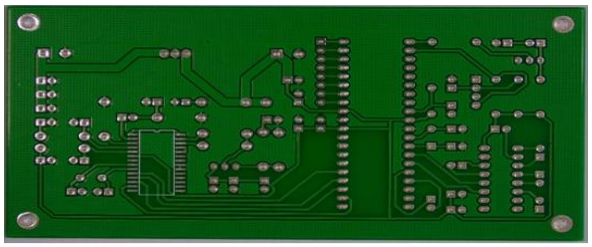


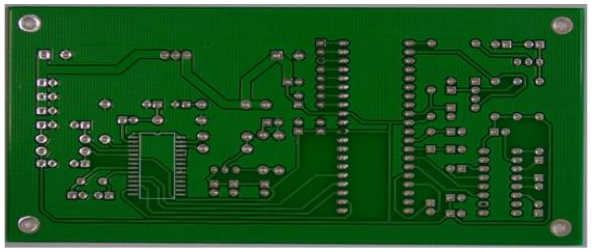
Figure. 1 Flowchart of the proposed hybrid deep learning model

Table 1. Data statistics of PCB defect dataset

Classes	Missing hole	Mouse bite	Open circuit	Short	Spur	Spurious copper
Total images	115	115	116	116	115	116
Number of defects	497	492	482	491	488	503



(a)



(b)

Figure. 2 (a) Original PCB defect image and (b) Ground truth image

defect classification: Bi-LSTM classifier. The flowchart of the proposed deep learning model is graphically stated in Fig. 1.

3.1 Dataset description

In this manuscript, the proposed model's effectiveness is tested on the PCB defect dataset. The PCB defect dataset consists of 693 PCB defective images with the pixel size of 2777×2138. The 693 PCB defective images have 2953 defects, which are categorized into six classes like mouse bite, spurious copper, short, spur, missing hole, and open circuit. The PCB defect dataset is available in the link of http://robotics.pkusz.edu.cn/resources/dataset/. Data statistics about PCB defect dataset is given in table 1 and the sample PCB defect image is represented in Fig. 2.

3.2 Feature extraction

After collecting the PCB defective images, the feature extraction is performed using BRISK and SURF. The BRISK descriptor is an effective texture descriptor that has better matching quality by generating valuable key-points in the collected images. The BRISK descriptor utilizes a symmetric sampling pattern over sample point of smooth pixels in the images, and then direction of each key point is identified that allows the orientation-normalized descriptor to achieve rotation invariance. Next, the BRISK key points are matched, where the intensity of the image is considered as p_i and the Gaussian smoothing with standard deviation is assumed as σ_i, which is directly proportional to the distance between circles and points. The key point k in the acquired images are patterned based on its position and scaling and the sampling point pairs are specified as (p_i, p_j). Additionally, the intensity of smooth value points are indicated as $X(p_i, \sigma_i)$ and $X(p_j, \sigma_j)$, which helps to determine the local gradients that is mathematically defined in Eq. (1).

$$G(p_i, p_j) = (p_j - p_i) \times \frac{X(p_j,\sigma_j) - X(p_i,\sigma_i)}{\|p_j - p_i\|^2} \quad (1)$$

Let us consider a set of all sampling-point pairs A, and then separate the pixel pairs into two sub-sets like short-distance pairs and long-distance pairs. The short-distance pairs is specified as S and the long-distance pairs is indicated as L, which are mathematically represented in the Eqs. (2) and (3).

$$S = \{(p_i, p_j) \epsilon A | \|p_j - p_i\| < \delta_{max}\} \subseteq A \quad (2)$$

$$L = \{(p_i, p_j) \epsilon A | \|p_j - p_i\| < \delta_{min}\} \subseteq A \quad (3)$$

Further, calculate the local gradient between long-distance pairs and the threshold distance is set to $\delta_{max} = 9.75t$ and $\delta_{min} = 9.75t$, where t is the scale of k. In this scenario, the point pairs are iterated using L for identifying the complete pattern direction of key points k that is mathematically defined in Eq. (4).

$$G = \begin{pmatrix} G_i \\ G_j \end{pmatrix} = \frac{1}{L} \times \sum_{(p_i,p_j)\epsilon L} G(p_i, p_j) \quad (4)$$

The sampling pattern rotation of orientation is stated as $\alpha = arctan\, 2(G_j, G_i)$ of the key points. The binary descriptor d_k is generated using short distance pairs, where each bit in d_k is calculated from a pair of S. In this study, the BRISK descriptor is 512 bit long and it is gathered by performing short distance intensity at every binary feature vectors b, which is mathematically represented in Eq. (5).

$$b = \begin{cases} 1, & X(p_j^\alpha, \sigma_j) > X(p_i^\alpha, \sigma_i) \\ 0, & otherwise \end{cases} \forall (p_x^\alpha, p_y^\alpha) \epsilon S \quad (5)$$

Where, b indicates 32 dimensional vectors at every key points.

Further, the SURF descriptor is a local feature descriptor, which is extensively employed in the image processing and computer vision fields. The SURF feature descriptor contains 3 steps such as interest point description, interest point localization, and integral image generation. In the SURF feature descriptor, the key points in the acquired image are detected based on scale space theory. The SURF feature descriptor uses Fast Hessian Detector to extract the SURF feature vectors from the acquired PCB defective images. The Hessian matrix is determined by using Eq. (6).

$$H(X,\sigma) = \begin{Bmatrix} C_{xx}(X,\sigma) & C_{xy}(X,\sigma) \\ C_{yx}(X,\sigma) & C_{yy}(X,\sigma) \end{Bmatrix} \quad (6)$$

Where, σ states scale, X represents points of the images, $C_{xx}(X,\sigma)$ states convolution of Gaussian 2nd order derivative $\frac{\partial^2}{\partial x^2} g(\sigma), g(\sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}}$ with the image in point X. Similarly, $C_{yy}(X,\sigma)$ and $C_{xy}(X,\sigma)$ states convolution of Gaussian 2nd order derivative $\frac{\partial^2}{\partial y^2} g(\sigma)\ and\ \frac{\partial^2}{\partial x \partial y} g(\sigma)$ with the image in point X.

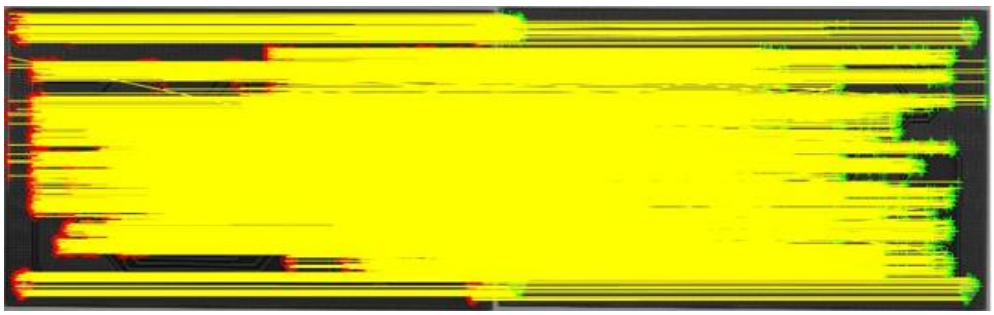
The box filter is used in the SURF feature descriptor as the approximation of the convolution Gaussian 2nd order derivative. The box filter is computed using the integral images for achieving convolution of box filters such as $B_{xx}, B_{xy}, and\ B_{yy}$. The approximate determinant of the Hessian matrix is determined for finding the image key-points, which is mathematically defined in Eq. (7).

$$Det\,[H(X,\sigma)] = B_{xx}B_{yy} - (0.912\,B_{xy})^2 \quad (7)$$

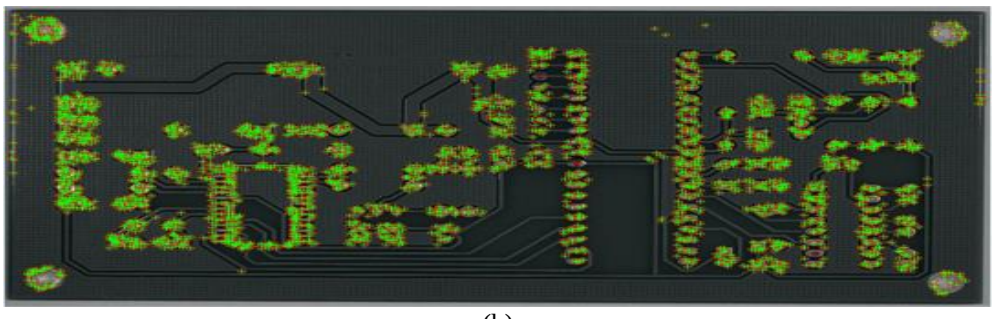
Where, 0.912 is used for stabling the determinant of Hessian matrix. The SURF descriptor utilizes the box filters to identify and match the interest points for obtaining the scale invariance. Whereas, the size of the box filters is altered for constructing the scale space and it is further portioned into octaves. Hence, the approximate determinant of the Hessian matrix is identified at a non-maximum suppression in $3 \times 3 \times 3$ neighborhood to find the maxima value. The SURF feature descriptor's scale σ and key-point's location are obtained with the reference to the maximum values. By finding the Haar Wavelet response, an orientation is assigned to the obtained key-points with the radius of $6s$, where s indicates sampling steps.

Next, an orientation direction is assigned to the key point's center with a square size of 20s. In addition, the square size is divided into 4×4 sub-regions, and it is again partitioned into 5×5 space points. Further, the vertical and horizontal Haar Wavelet response dx and dy are determined at every space points. By using Haar Wavelet response, each sub-regions creates four-dimensional vector that is mathematically determined in Eq. (8).

$$v = (\sum dx\,, \sum dy\,, \sum|dx|, \sum|dy|) \quad (8)$$



(a)



(b)

Figure. 3: (a) Extracted features and (b) feature matching

Lastly, all the sub-regions are merged into vectors $4 \times (4 \times 4)$ that result in 64 dimensional vectors at every key points, which are further used to accomplish the matching process. In the next step, the dimension of the extracted feature vectors is reduced by employing stacked autoencoder that effectively reduces the computational complexity and running time of the classifier. Whereas, the extracted features and feature matching are graphically depicted in Fig. 3.

3.3 Dimensionality reduction

After feature extraction, the feature dimensionality reduction is carried out by utilizing stacked autoencoder. The stacked autoencoder consistently performed well in the dimensionality reduction compared to conventional classifiers. The stacked autoencoder is the feed forward neural network, which comprises of multiple hidden layers, an input layer and an output layer, which are described in the Eqs. (9) and (10).

$$Z^{(l)} = y^{(l-1)}W^{(l)} + b^{(l)} \tag{9}$$

$$y^{(l)} = g(Z^{(l)}) \tag{10}$$

Where, $Z^{(l)}$ indicates pre-activation layer of vector l, $y^{(l-1)}$ represents input of present layer l and output of previous layers $l-1$, $W^{(l)} \in \mathbb{R}^{n_i \times n_0}$ denotes matrix of learnable biases b, $y^{(l)}$ represents input of the model, $g(.)$ indicates nonlinear activation function, $l \in [1, \dots L]$ denotes l^{th} layer, and $y^{(L)}$ indicates final layer output. In this scenario, the ReLU is utilized as an activation function that superiorly enhances the computer efficiency, and increases the learning rate of the model for better dimensionality reduction. Additionally, the softmax non-linearity function is applied in the output layer to obtain better probability interpretation that is mathematically stated in Eq. (11).

$$softmax\left(Z^{(L)}\right) = \frac{expZ_k}{\sum_{k=1}^{K} expZ_k} \tag{11}$$

Where, K denotes output classes. In stacked autoencoder, the cross entropy loss function is applied to deal with the optimization problem, which is mathematically determined in Eq. (12).

$$C = -\sum_{k=1}^{K} \hat{y}_k \log(y_k^{(L)}) \tag{12}$$

Where, $y^{(L)}$ specifies model output and $\hat{y}_k \in \{0,1\}^k$ represents encoded label. In this manuscript,

Figure. 4 Defect detected PCB image

the stacked autoencoder is designed for learning the low dimensional feature vectors. The stacked autoencoder with a hidden layer is mathematically determined in Eq. (13).

$$h_e = a_1(W_e x) \; and \; \hat{x} = a_2(W_d \, h_e), \tag{13}$$

Where , x indicates input feature vectors and $\hat{x}$ represents reconstructed feature vectors, W_e and W_d are matrices, which indicates the linear combination of inputs for encoding and decoding sections. Further, a_1 and a_2 are constant values, and h_e is the output of bottleneck layer and it is considered as the low dimensional representation of the input feature vectors. The parameter setting of stacked autoencoder is indicated as follows: sparsity regularization is 4, L2 weight regularization is 0.004, sparsity proportion is 0.150, number of hidden layers is 100, maximum iterations: SAE learning is 100, and maximum iterations: softmax learning is 100. The defect detected PCB image is denoted in Fig. 4.

3.4 Classification

The optimized 26 dimensional vectors are given as the input to the Bi-LSTM network for classifying the six PCB defects such as mouse bite, spurious copper, short, spur, missing hole, and open circuit. The LSTM network is the advanced version of the recurrent neural network that uses memory cell for controlling input, output and forget gates, and storing the network temporal state. The input and output gates are used for controlling the input and output flows of the memory cell. Additionally, the forget gate is attached to the memory cell that transfers output information from current neuron to the next neuron. The information is stored in memory cell, if the input has high activation, and the information is passed to the next neuron, if the output has high activation. The LSTM network contains input gate i_t, forget gate f_t, cell c_t and output gate o_t, which are indicated in the Eqs. (14) to (17).

$$i_t = \sigma(W_{ih} h_{t-1} + W_{ia} a_t + b_i) \tag{14}$$

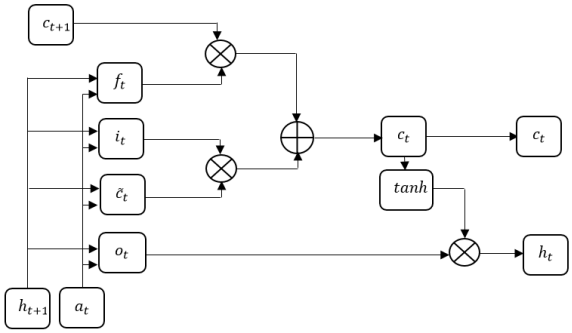


Figure. 5 Architecture of LSTM unit

$$f_t = \sigma(W_{fh}h_{t-1} + W_{fa}a_t + b_f) \quad (15)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(W_{ch}h_{t-1} + W_{ca}a_t + b_c) \quad (16)$$

$$o_t = \sigma(W_{oh}h_{t-1} + W_{oa}a_t + b_0) \quad (17)$$

Where, $a_t = A[t,.] \in \mathbb{R}^F$ indicates quasi-periodic feature, $\odot$states point-wise multiplication, $\sigma(.)$ states sigmoid activation function, $tanh\,(.)$ represents hyperbolic tangent activation function, h_{t-1} denotes output of the prior LSTM unit, W and b indicates work coefficients. The output of LSTM unit h_t is mathematically defined in Eq. (18), and graphically depicted in Fig. 5.

$$h_t = o_t \odot tanh(c_t) \quad (18)$$

As indicated in Fig. 5, the term h_t includes information of the prior time steps t of a cell c_t and output gate o_t. The cell states $\{c_t|t = 1,2,..T\}$ learns the memory information of the temporal quasi-periodic feature vectors for short and long time period based on dependency relation during data training. The Bi-LSTM network is developed to overcome the problems of LSTM cell, where it effectively works on the prior content, but it cannot utilize the future one. In Bi-LSTM network, the input flows in two direction that makes this network is different from the conventional LSTM network. Hence, the LSTM network makes flow in one direction (neither backward nor forward), where Bi-LSTM network makes input flow in both directions for preserving the past and the future information. The parameter setting of Bi-LSTM network is given as follows: minimum batch size is 27, maximum epochs is 200, gradient threshold is one and the execution environment is graphics processing units.

4. Experimental results

The proposed hybrid deep learning model (stacked autoencoder with Bi-LSTM) performance is tested using MATLAB 2020 software environment on a system configuration with windows 10 operating system, 16 GB random access memory, 4 TB hard disk, and Intel core i7 processor. The proposed hybrid deep learning model's efficiency is validated utilizing five performance measures: F1-score, MCC, specificity, classification accuracy, and sensitivity. The classification accuracy is an important performance measure in the PCB defect detection that denotes how close the achieved result to the true values. Correspondingly, the specificity and sensitivity test measures are used for identifying the features without and with the PCB defects. The mathematical representation of the classification accuracy, specificity, and sensitivity are represented in the Eqs. (19) to (21). Where, FP, FN, TP, and TN indicates false positive, false negative, true positive, and true negative.

Table 2. Confusion matrix related to the PCB defect detection

		Actual class	
		Defects	**Non-defects**
Predicted class	**Defects**	**True positive**	**False positive**
	Non-defects	False negative	True negative

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (19)$$

$$Specificity = \frac{TN}{TN+FP} \times 100 \quad (20)$$

$$Sensitivity = \frac{TP}{TP+FN} \times 100 \quad (21)$$

In addition to this, the F1-score is the harmonic mean of sensitivity and precision values, and the parametric value of MCC lies between 1 to -1, where the proposed hybrid deep learning model is effective in the PCB defect detection, when the parametric value is 1. The mathematical formula of MCC and F1-score is indicated in the Eqs. (22) and (23). Further, the confusion matrix of this research is depicted in Table 2.

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \times 100 \quad (22)$$

$$F1 - score = \frac{2TP}{FP+2TP+FN} \times 100 \quad (23)$$

4.1 Quantitative evaluation

In this manuscript, the proposed hybrid deep learning model performance is evaluated on the PCB

defect dataset, which comprises of 693 PCB defective images in that 80:20% of the defective images is applied for model training and testing. In addition to this, the 10-fold cross validation is performed on the data that reduces the variance of the resulting estimation and computational time. By inspecting table 3, the performance evaluation is done by varying the classification techniques: random forest, K-Nearest Neighbour (KNN), decision tree, and Bi-LSTM network along with and without the stacked autoencoder technique. As represented in the table 3, the combination: stacked autoencoder with Bi-LSTM network achieved high performance in the PCB defect detection with accuracy of 99.99%, F1-score of 99.41%, sensitivity of 98.35%, specificity of 99.67%, and MCC of 98.89% on the PCB defect dataset. The obtained experimental result is better compared to other individual classification techniques such as random forest, KNN, and decision tree. Related to the comparative classification techniques, the Bi-LSTM classifier significantly improves the model's performance on sequence classification problems. The experimental results of different classifiers with and without using stacked autoencoder technique is graphically indicated in the Fig. 6 and 7.

By viewing Table 4, the stacked autoencoder technique obtained high classification performance with Bi-LSTM classifier compared to the conventional techniques: Principal Component Analysis (PCA), Probabilistic PCA (PPCA), Reconstruction PCA (RPCA), and Reconstruction Independent Component Analysis (RICA). The stacked autoencoder technique is a good choice for the feature dimensionality reduction and visualization of dataset with a larger number of the variables. Hence, the comparison results of the different feature dimensionality reduction techniques with Bi-LSTM classifier is depicted in Fig. 8.

Table 3. Experimental results of the proposed model by varying the classifiers

Without stacked autoencoder					
Classifiers	**Accuracy (%)**	**Sensitivity (%)**	**Specificity (%)**	**MCC (%)**	**F1-score (%)**
KNN	80.33	82.42	81.77	82.36	83.75
Random forest	63.73	64.54	64.25	65.42	63.38
Decision tree	88.38	86.48	87.74	88.40	87.51
Bi-LSTM	95.79	96.67	94.59	95.87	95.75
With stacked autoencoder					
Classifiers	Accuracy (%)	Sensitivity (%)	Specificity (%)	MCC (%)	F1-score (%)
KNN	92.50	92.32	91.75	92.67	93.74
Random forest	73.74	74.51	73.27	72.41	73.53
Decision tree	97.33	96.54	96.77	96.30	97.58
Bi-LSTM	99.99	98.35	99.67	98.89	99.41

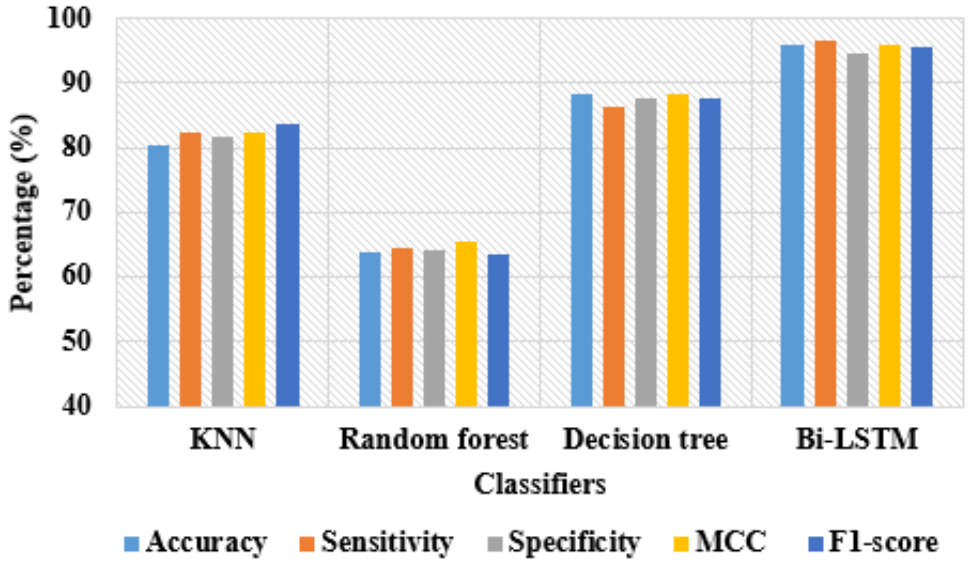


Figure. 6 Comparison results of different classifiers without using stacked autoencoder

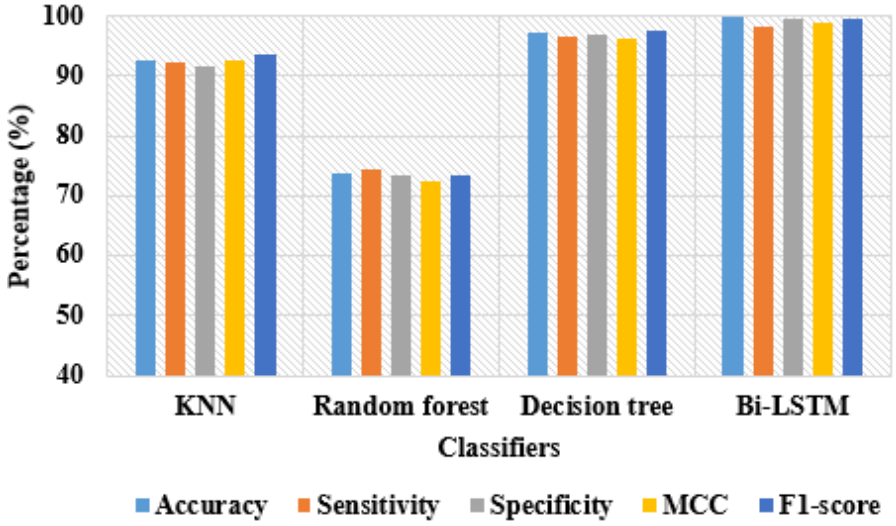


Figure. 7 Comparison results of different classifiers with using stacked autoencoder

Table 4. Experimental results of the proposed model by varying the feature dimensionality reduction techniques

Techniques	Accuracy (%)	Sensitivity (%)	Specificity (%)	MCC (%)	F1-score (%)
PCA	34.17	34.69	31.67	35.32	36.83
PPCA	47.50	45.50	44.74	48.38	47.84
RICA	23.33	25.30	24.28	23.49	23.86
RPCA	40.83	41.93	40.74	39.75	40.33
Stacked autoencoder	99.99	98.35	99.67	98.89	99.41

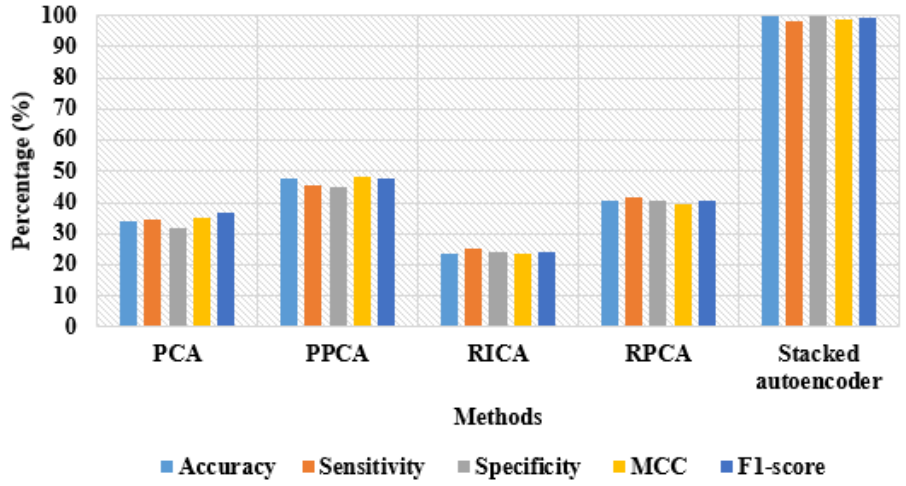


Figure. 8 Comparison results of different feature dimensionality reduction techniques with Bi-LSTM classifier

4.2 Comparative evaluation

The investigation between the proposed hybrid deep learning model and the comparative models is depicted in Table 5. J. Kim [15] developed a novel automated PCB defect detection model based on skip connected convolutional autoencoder. The quantitative evaluation on the PCB defect dataset showed that the presented deep learning model attained better defect detection performance with F1-score of 98.01% and classification accuracy of 98.08%. Correspondingly, R. Ding [20] integrated both k-means clustering algorithm and TDD-Net model for an effective PCB defect detection by strengthening the relationship of the extracted feature map. The experimental analysis showed that the developed deep learning model achieved a mean average precision of 98.90% on the PCB defect dataset.

Related to these existing papers, the proposed hybrid deep learning model obtained high performance in the PCB defect detection. The proposed hybrid deep learning model evenly distributed the extracted multi-dimensional features by employing stacked autoencoder and the Bi-LSTM

Table 5. Comparative valuation between the proposed and the existing models

Models	F1-score (%)	Accuracy (%)	Mean average precision (%)
Skip connected convolutional autoencoder [15]	98.01	98.08	-
TDD-Net model [20]	-	-	98.90
Hybrid deep learning model	99.41	99.99	99.54

network is a quick and effective learner for the defect type classification, which are the main problems mentioned in the literature section.

5. Conclusion

In this manuscript, a new hybrid deep learning model (stacked autoencoder with Bi-LSTM) is introduced for an effective PCB defect detection. After collecting the PCB defective images, the feature extraction is accomplished by using BRISK and SURF descriptors for extracting the global feature vectors from the PCB defective images that superiorly reduces the semantic space between the feature sub-sets. Next, the stacked autoencoder technique is developed to reduce the dimension of the extracted feature values that diminishes the system complexity to linear$O(N)$, where O states order of magnitude, and N states size of the data. After feature dimensionality reduction, the defect type classification is accomplished by using Bi-LSTM classifier for classifying the six defect types like mouse bite, spurious copper, short, spur, missing hole, and open circuit. The proposed hybrid deep learning model attained 99.99% of classification accuracy, 98.35% of sensitivity, 99.67% of specificity, 98.89% of MCC, and 99.41% of F1-score in PCB defect detection, where the achieved results are better related to the comparative dimensionality reduction and classification techniques. As a future extension, a new hyper-parameter optimization algorithm is included in the Bi-LSTM network to further enhance the PCB defect detection.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

The paper conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation, writing—review and editing, visualization, have been done by 1st author. The supervision and project administration, have been done by 2nd author.

Parameters	Notation
p_i	Intensity of the image
σ_i	Gaussian smoothing with standard deviation
(p_i, p_j)	Sampling point pairs
S	Short-distance pairs
L	Long-distance pairs
b	32 dimensional vectors at every key points
σ	Scale
X	Points of the images
dx and dy	Vertical and horizontal Haar Wavelet response
$Z^{(l)}$	Pre-activation layer of vector l
$y^{(l-1)}$	Input of present layer l and output of previous layers $l-1$
$W^{(l)} \in \mathbb{R}^{n_i \times n_0}$	Matrix of learnable biases
$g(.)$	Nonlinear activation function
$y^{(L)}$	Final layer output
K	Output classes
h_e	Output of bottleneck layer
i_t	Input gate
f_t	Forget gate
c_t	Cell state
o_t	Output gate
$a_t = A[t,.] \in \mathbb{R}^F$	Quasi-periodic feature
$\odot$	Point-wise multiplication
$\sigma(.)$	Sigmoid activation function
$tanh\,(.)$	Hyperbolic tangent activation function
FP, FN, TP, TN	False positive, false negative, true positive, and true negative

References

[1] L. Zhang, Y. Jin, X. Yang, X. Li, X. Duan, Y. Sun, and H. Liu, "Convolutional neural network-based multi-label classification of PCB defects", *The Journal of Engineering*, Vol. 2018, pp. 1612-1616, 2018.

[2] C. Zhang, W. Shi, X. Li, H. Zhang, and H. Liu, "Improved bare PCB defect detection approach based on deep feature learning", *The Journal of Engineering*, Vol. 2018, pp. 1415-1420, 2018.

[3] Z. Lu, Q. He, X. Xiang, and H. Liu, "Defect detection of PCB based on Bayes feature fusion",

The Journal of Engineering, Vol. 2018, pp. 1741-1745, 2018.

[4] M. H. Annaby, Y. M. Fouda, and M. A. Rushdi, "Improved normalized cross-correlation for defect detection in printed-circuit boards", *IEEE Transactions on Semiconductor Manufacturing*, Vol. 32, pp. 199-211, 2019.

[5] W. Huang, G. Hua, Z. Yu, and H. Liu, "Recurrent spatial transformer network for high-accuracy image registration in moving PCB defect detection", *The Journal of Engineering*, Vol. 2020, pp. 438-443, 2020.

[6] A. Mujeeb, W. Dai, M. Erdt, and A. Sourin, "One class based feature learning approach for defect detection using deep autoencoders", *Advanced Engineering Informatics*, Vol. 42, pp. 100933, 2019.

[7] Y. T. Li, P. Kuo, and J. I. Guo, "Automatic Industry PCB Board DIP Process Defect Detection System Based on Deep Ensemble Self-Adaption Method", *IEEE Transactions on Components, Packaging and Manufacturing Technology*, Vol. 11, pp. 312-323, 2020.

[8] P. Wei, C. Liu, M. Liu, Y. Gao, and H. Liu, "CNN-based reference comparison method for classifying bare PCB defects", *The Journal of Engineering*, Vol. 2018, pp. 1528-1533, 2018.

[9] H. Xie, Y. Li, X. Li, and L. He, "A Method for Surface Defect Detection of Printed Circuit Board Based on Improved YOLOv4", In: *Proc. of 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering*, pp. 851-857, 2021.

[10] I. Volkau, A. Mujeeb, D. Wenting, E. Marius, and S. Alexei, "Detection defect in printed circuit boards using unsupervised feature extraction upon transfer learning", In: *Proc. of 2019 International Conference on Cyberworlds*, pp. 101-108, 2019.

[11] J. Li, J. Gu, Z. Huang, and J. Wen, "Application research of improved YOLO V3 algorithm in PCB electronic component detection", *Applied Sciences*, Vol. 9, p. 3750, 2019.

[12] B. Ghosh, M. K. Bhuyan, P. Sasmal, Y. Iwahori, and P. Gadde, "Defect classification of printed circuit boards based on transfer learning", *IEEE Applied Signal Processing Conference*, pp. 245-248, 2018.

[13] D. M. Tsai and P. H. Jen, "Autoencoder-based anomaly detection for surface defect inspection", *Advanced Engineering Informatics*, Vol. 48, p. 101272, 2021.

[14] A. A. I. Hassanin, F. E. A. E. Samie, and G. M. E. Banby, "A real-time approach for automatic defect detection from PCBs based on SURF features and morphological operations", *Multimedia Tools and Applications*, Vol. 78, pp. 34437-34457, 2019.

[15] J. Kim, J. Ko, H. Choi, and H. Kim, "Printed circuit board defect detection using deep learning via a skip-connected convolutional autoencoder", *Sensors*, Vol. 21, p. 4968, 2021.

[16] V. A. Adibhatla, H. C. Chih, C. C. Hsu, J. Cheng, M. F. Abbod, and J. S. Shieh, "Defect detection in printed circuit boards using you-only-look-once convolutional neural networks", *Electronics*, Vol. 9, p. 1547, 2020.

[17] B. Hu and J. Wang, "Detection of PCB surface defects with improved faster-RCNN and feature pyramid network", *IEEE Access*, Vol. 8, pp. 108335-108345, 2020.

[18] V. H. Gaidhane, Y. V. Hote, and V. Singh, "An efficient similarity measure approach for PCB surface defect detection", *Pattern Analysis and Applications*, Vol. 21, pp. 277-289, 2018.

[19] E. H. Yuk, S. H. Park, C. S. Park, and J. G. Baek, "Feature-learning-based printed circuit board inspection via speeded-up robust features and random forest", *Applied Sciences*, Vol. 8, p. 932, 2018.

[20] R. Ding, L. Dai, G. Li, and H. Liu, "TDD-net: a tiny defect detection network for printed circuit boards", *CAAI Transactions on Intelligence Technology*, Vol. 4, pp. 110-116, 2019.