Received: October 19, 2024. Revised: December 4, 2024.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Bidirectional Long Short-Term Memory with Parametric Flatten-T Swish Activation Function for Land Use Land Cover Classification

Woothukadu Thirumaran Chembian[1]* Kandasamy Thinakaran[2] Krishna Murthi Sankar[3] Marimuthu Venkatesan[4] Ethala Kamalanaban[1]

[1]*Department of Computer Science and Engineering, Vel Tech High Tech Dr.Rangarajan Dr.Sakunthala Engineering College (Autonomous), Chennai, India*
[2]*Department of Computer Science and Engineering, Saveetha Institute of Medical and Technical Sciences, Saveetha University, Chennai, India*
[3]*Department of Computer Science and Engineering - Emerging Technologies, CVR College of Engineering, Ibrahimpatnam, India*
[4]*Department of Artificial Intelligence and Data Science (AI&DS), Panimalar Engineering College, Chennai, India*
* Corresponding author's Email: wtchembian@velhightech.com

Abstract: The Land Use Land Cover (LULC) classification in aerial images involves analysing and extracting data to accurately classify various land cover types including urban areas, forests and agricultural fields. However, high dimensional feature space makes this task challenging, which leads to reduced classification performance. Therefore, Personal local search memory-based Woodpecker Mating Algorithm and Bidirectional Long Short-Term Memory with Parametric Flatten-T Swish (PWMA-BiLSTM with PFTS) are proposed in this research for LULC classification. The PWMA efficiently integrates global search capabilities with personal local search memory to enhance the exploration of the feature set. The BiLSTM captures complex and temporal patterns within the selected features through its bidirectional processing ability, which allows the capture of contextual data in LULC classification. The performance measures like accuracy, sensitivity, precision, f1-score, specificity and computational time are used to evaluate the performance of PWMA-BiLSTM with PFTS. The PWMA-BiLSTM with PFTS achieves an accuracy of 99.95% and 99.87% for Aerial Image (AID) and UC Merced (UCM) datasets, which outperforms Fully Convolutional Network (FCN).

Keywords: Bidirectional long short-term memory, Land use land cover, Parametric flatten-T swish, Personal local search memory, Woodpecker Mating Algorithm.

1. Introduction

Landscape changes provide significant information about the effects of various land use patterns and types on the environment. The aerial image monitoring used to examine affected landscapes is very helpful [1, 2]. In the Land Use Land Cover (LULC) classification, labels designate physical land type and how the region is utilised [3]. The remote sensing images are analyzed using various spatial, temporal and spectral solutions, in which the accuracy is primarily affected by spatial resolution [4]. The rapid increase in number of remote sensing images and highly complex spatial patterns and geometric structures are significant to understanding semantic content [5, 6]. Land use identification denotes to classifying lands based on its use and purpose, which ensures land is utilised with local region regulations and other land use strategies [7-9]. The land use types include commercial, industrial, residential, recreational and agricultural [10]. Application areas such as environmental pollution monitoring, LULC mapping and precise agriculture require images with high spatial and spectral resolution for rapid analysis and higher accuracy [11].

The recent researches in this LULC classification are attained through higher-resolution aerial images of land surface by drones, satellites and other devices [12]. However, the present image processing techniques are developed for medium-resolution sensors and a smaller number of bands, which makes them inadequate for handling higher spectral-resolution images [13, 14]. This issue reduces the effectiveness of conventional techniques according to the analysis of spectral features [15]. To solve this issue and obtain better classification accuracy, complementary information such as texture data is significant to model spatial pixel characteristics and is required to integrate the process of higher-resolution images [16]. The existing LULC classification has less convergence speed, thereby leading to less detection performance because it does not learn optimum solutions. Moreover, it is a challenging task because of the high dimensional feature space, which reduces the classification performance. The major contribution of this research is as below:

- The PWMA enhances the feature selection in LULC classification by balancing local exploitation and global exploration. Moreover, it can identify patterns and is robust to irrelevant and noisy features which leads to reliable feature selection.
- The BiLSTM captures complex and temporal patterns present in the selected features and it has an ability of bidirectional processing thereby capturing contextual data in LULC for differentiating classes.
- The PFTS enhances the non-linear, smooth transformation by tunable parameters thereby leading to better generalization and quick convergence in LULC classification. Using BiLSTM with PFTS reduces the computation time when maintaining higher classification performance.

This research paper is prepared as follows: Section 2 analyses the literature and Section 3 describes the proposed methodology. Section 4 illustrates result evaluation and the conclusion of this research is given in Section 5.

2. Literature review

Recently, Deep Learning (DL) techniques were extensively applied for LULC classification because of their efficiency which was reviewed to understand the performance.

Hilal [17] developed a Deep Transfer Learning - based Fusion model for Environmental Remote-Sensing Image Classification (DTLF-ERSIC). The DTLF-ERSIC includes entropy-based fusion methods such as Discrete Local Binary Pattern (DLBP), Residual Network (ResNet50), and EfficientNet models. Furthermore, it integrates numerous feature vectors to obtain enhanced classification performance. However, it was unable to achieve effective feature fusion due to the presence of highly correlated features.

Ghadi [18] suggested an FCN for scene classification. The fuzzy c-means segmentation method was applied to classify different objects and it was labelled by Markov Random Field (MRF). Then, CNN features were extracted and integrated to classify objects. Lastly, it was transformed to FCN for scene classification with triplet relationships. The FCN effectively captures structured data and ensures generated sample quality. However, it has less convergence speed which leads to less detection performance because it unable to learn optimal solutions.

Hilal [19] introduced Fuzzy Cognitive Maps with Bird Swarm algorithm-based Remote Sensing Image Classification (FCMBS-RSIC). The FCM was used to allocate accurate class labels and classifier performance was enhanced through Bird Swarm Algorithm (BSA). The preprocessing was used to transform the sentinel image into the appropriate form and the RetinaNet model was applied to produce features. The FCMBS-RSIC has high classification accuracy. However, it has less classification accuracy because of gradient issues.

Li [20] presented a remote sensing scene classification through contrastive vision-language supervision known as RS-CLIP. The Rs-CLIP was able to learn semantic visual representation. The curriculum learning strategy was applied to produce labels automatically and enhance the zero-shot remote sensing classification with numerous stages. However, the overfitting issue occurred due to not considering feature selection thereby reducing classification performance.

Noppitak and Surinta [21] implemented an Ensemble Convolution Neural Network (CNN) for land use classification. The dropCyclic was applied to minimize learning rate and discover local optima in a following cycle. The dropcyclic utilized three CNN architectures as backbone such as VGG16, VGG19 and MobileNetV2. However, it has higher dimensional features thereby reducing classification performance.

Ningbo Guo [22] suggested a dual-model architecture with multilevel feature fusion named as XE-Net for land use classification. The high, low and middle level features are extracted through EfficientNEt-V2 and Xception by transfer learning

model. The triple-level features are extracted by dual models based on similar level. The XE-Net improves the three-scale features which enhances the discriminative features. Lastly, the discriminative features are given as input to XE-Net to differentiate various lands and achieve high accuracy. However, the relevant features are not selected through feature selection therefore the model performance is reduced.

From the above analysis, the existing techniques have drawbacks such as gradient issues, and higher dimensional features thereby reducing classification performance. Moreover, it has an inability for feature fusion due to highly correlated features. It has less convergence speed because it unable to learn optimal solutions. The overfitting issue occurred due to not considering feature selection thereby reducing classification performance. To overcome this problem, the PWMA-BiLSTM with PFTS is proposed for LULC classification. The PWMA can search patterns and is robust to irrelevant and noisy features which leads to reliable feature selection. The BiLSTM captures complex and temporal patterns present in the selected features which leads to informative presentation in LULC for differentiating classes.

3. Proposed method

The PWMA-BiLSTM with PFTS is proposed in this research for LULC classification. The Histogram of Oriented Gradient (HOG), Local Ternary Pattern (LTP) and Convolutional Neural Network (CNN) based feature extractions are used to extract gradient, texture and hierarchical features from AID and UCM datasets. Then, the PWMA is used for feature selection which selects relevant features and is classified by BiLSTM with PFTS method. Fig. 1 shows an overall process of the proposed PWMA-BiLSTM with PFTS in LULC classification.

3.1 Dataset

The datasets used in this research are AID and UCM which are explained in the following subsections with the number of classes, number of images, image sizes and its spatial resolutions.

AID dataset: The AID dataset [23] is gathered from Google Earth imagery and it has 30 classes and 10000 images in which every class has 200-400 images. The image size is 600×600 and its spatial resolution is 0.5-8m. Fig. 2 shows the sample images.

UCM dataset: The UCM dataset [24] is gathered from the University of California at Merced lab and it has 21 classes in that each class has 100 images totally of 2100 images. The image size is 256×256 and its spatial resolution is 0.3m. Fig. 3 shows the sample images.

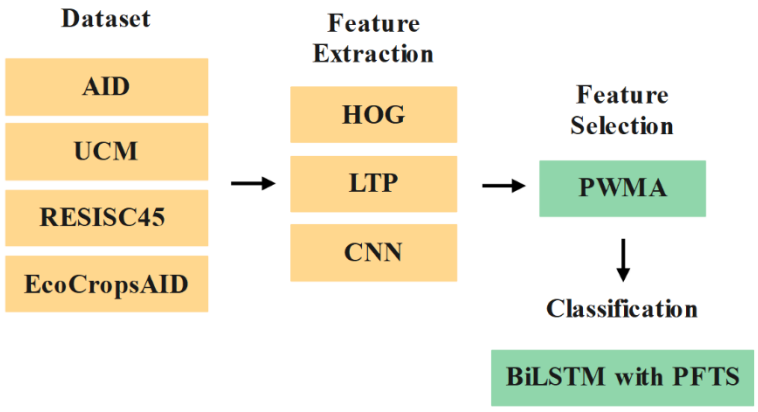

Figure. 1 Overall process of proposed approach in LULC classification

Figure. 2 Sample dataset images

Figure. 3 Sample dataset images

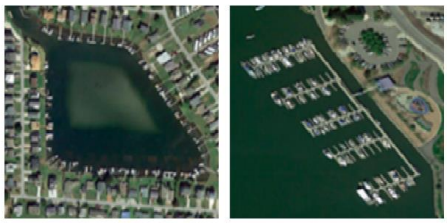

Figure. 4 Sample dataset images

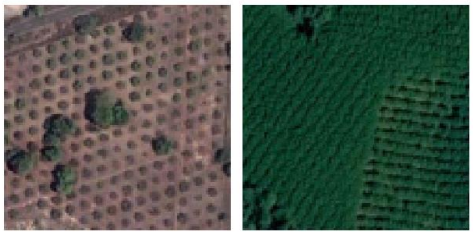

Figure. 5 Sample dataset images

RESISE45 dataset: The RESISE45 dataset [18] contains 31,500 scene images of 45 classes. Every class includes 700 images in which every class has minimum 2 and maximum 10 objects. Fig. 4 shows the sample images.

EcoCropsAID dataset: The EcoCropsAID dataset [21] is collected from Google Earth application through resolution of 30-0.2 meters. The images are stored with 600×600 pixels in RGB format. It has 5,400 images of 5 classes. Fig. 5 shows the sample images.

3.2 Feature extraction

The raw images from datasets are provided for the feature extraction process to extract gradient, texture and hierarchical features. The HOG, LTP and CNN based feature extractions are used in this research which are explained as follows.

3.2.1. HOG

The gradients and edges are extracted from HOG which are required to present structures and shapes of objects in LULC images. The HOG captures the magnitude and orientation of intensity variations in the image by extracting boundaries and edges between different objects which is useful to differentiate different LULC images. The HOG is based on gathering gradient directions by small spatial pixels named cells. The local 1D histogram of edge gradients or orientations is collected through HOG for every cell [25]. This procedure is performed by producing a local histogram of high spatial regions which have cells and applying results to normalize each cell in the block.

3.2.2. LTP

The texture-based features are extracted from LTP which captures spatial intensities in images and captures patterns such as smoothness and roughness since different land have various textual patterns. The LTP has a 3-scored code and texture operator s is robust to noise [25] [26] which is stated in Eq. (1).

$$s(x) = \begin{cases} 2, & if \ x \geq t, \\ 1, & if \ |x| < t, \\ 0, & if \ x \leq t, \end{cases} \quad (1)$$

Where, t is a user threshold. After the threshold step, the lower and upper patterns are constructed and coded. The lower and upper pattern code concatenation is an LTP operator.

3.2.3. CNN based feature extraction

By using CNN-based models such as ResNet50, the hierarchical deep features are extracted and it is capable of learning intricate patterns. The ResNet50 utilizes residual blocks to solve the gradient degradation and disappearance challenges in CNN [17]. The residual block in ResNet stimulates the residuals with input and output of its residual blocks. The residual function equation is stated in Eq. (2).

$$y = F(x, W) + x \quad (2)$$

Where, x, W and y are residual block input, weight and output. The ResNet includes different residual blocks in which convolutional kernel size of convolution layer is different. The features from HOG, LTP and CNN are concatenated by using Eq. (3).

$$x = \{HOG, LTP, CNN\} \quad (3)$$

3.3 Feature selection

The extracted features are provided to PWMA-based feature selection for selecting features thereby enhancing the classification performance. The extracted features have high dimensional features that reduce the model complexity thereby reducing classification performance. Therefore, the feature selection is significant which reduces the high dimensional features by selecting relevant features and enhancing the classification performance. The WMA is simulated by the mating behavior of woodpeckers and its major metaphor is the drumming sound of male woodpeckers to attract the female woodpeckers. The WMA has dual groups such as

male and female woodpeckers. The male is the best location explored through them whereas the female is a major search factor. Female woodpeckers are attracted by drumming sounds caused by males. The attraction level is based on received sound quality that is directly proportional to sound intensity. This sound quality is stated in Eq. (4),

$$I = \frac{P_S}{4\pi r^2} \tag{4}$$

Where, P_S is a sound wave energy at sound source and r is a Euclidean distance hearer to sound source. In traditional WMA, the search agent updates their location according to its male woodpecker's location and population of best males. In proposed algorithm, a piece of local memory is concentrated for the best personal experience storage of each woodpecker. This personal memory is applied to update the location of its respective woodpecker as given in Eq. (5).

$$x_i^{t+1} = x_i^t + \left[r_1 \times \frac{\delta_i^t \times \left(\alpha_{gpop} \times (x_{gpop}^t - x_i^t) + \alpha_{mj} \times \left(x_{mj}^t - x_i^t \right) \right)}{2} + r_2 \times \left(\delta_i^t \times \left(x_{ppop}^t - x_i^t \right) \right) \right] \tag{5}$$

Where, x_i^t and δ_i^t are present position and random vector of woodpecker i in tth iteration, r_1 is a random number from normal distribution within a range of $[0, 1]$. x_{gpop}^t is a best member position of population, x_{mj}^t is a jth male woodpecker, x_{ppop}^t is a location of memory. By using personal local search memory abilities, it enforces the accurate approximation of global search optimum. In WMA, every female updates their location as affected through drumming from best population member and male one has shortest distance from it. The self-tuned value at the iteration cycle is stated in Eq. (6), parameter α $(\alpha_{gpop}, \alpha_{mj})$ are attained through Eq. (7).

$$\delta_i^t = r_3 \times Tansig\left(1 - \frac{t}{t_{max}}\right) \tag{6}$$

$$\alpha = \frac{1}{1 + SI_j^i} \tag{7}$$

Where, r_3 is a random number from normal distribution within a range of $[0, 3]$, $Tansig$ is a tangent sigmoid function, t and t_{max} are number of present and maximum iterations. If $\delta > 1$, search factor diverges from target point that leads exploration. If $\delta \leq 1$, female woodpecker convergence to male thereby leading exploitation. The α is an attractiveness of jth male to ith female woodpecker, SI_j^i is a sound intensity target which is heard by female. Moreover, α is a female woodpecker step size which denotes particularly how adjacent it reaches its respective male woodpecker. In WMA, the Running Away (RA) function is taken as stochastic movement factors once drumming sounds are overloaded and woodpecker is attracted. This function comprises dual operators such as Random RA (RRA) and Gpop RA (GRA). Every woodpecker in each generation is directly proportional to its respective α score based on Eq. (8) and the $H\alpha$ value is estimated in the initial iteration by Eq. (9).

$$RA = \begin{cases} RRA & if\ \alpha \geq H\alpha \\ GRA & else \end{cases} \tag{8}$$

$$H\alpha = 0.8 \times \frac{\sum_1^{n-1} \alpha_{gpop}^i}{N-1} \tag{9}$$

Where, N is a woodpeckers population size, $H\alpha$ is a highest level of α. The RRA is generally stochastic moves among search space which is concentrated on exploration directly and implemented by Eq. (10).

$$x_{RRA}^i = lb - (lb - ub) \times r_4 \tag{10}$$

Where, x_{RRA}^i is a new element position attained from RRA on ith woodpecker, lb and ub are lower and upper bound variables, r_4 is a random number from normal distribution within a range of $[0, 1]$. The GRA operator leads stochastic changes in a few variables which is regarding female woodpeckers according to best male and location of random other woodpecker. The GRA operation is implemented by Eq. (11).

$$x_{GRA}^i = x_i^t + GRA_{bit} \times \left\{ \left(x_{gpop}^t - x_r \right) \times R \right\} \tag{11}$$

Where, x_{GRA}^i is a new element location attained from GRA on i th woodpecker, x_i^t is a random woodpecker location, GRA_{bit} is a binary vector, x_{gpop}^t is a location of best woodpecker over the population, x_r is a random woodpecker location, R is a random numbers from random distribution within a range of $[-1, 1]$. Fig. 6 denotes the flowchart of PWMA.

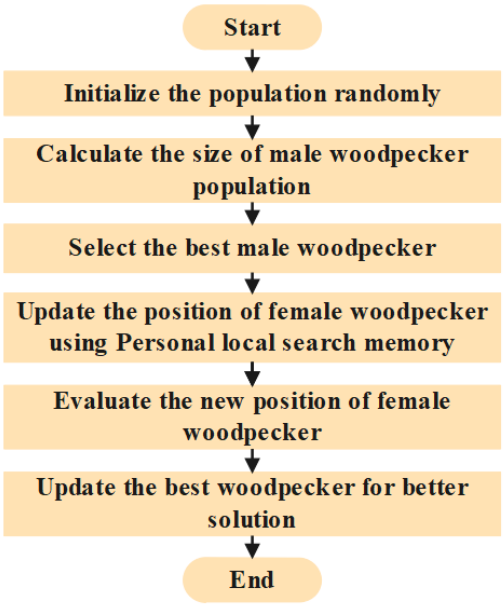

Figure. 6 Flowchart of PWMA

The maximum accuracy is considered as fitness function in this research. Based on this fitness function, the PWMA selects the best features and the fitness formula is stated in Eq. (12).

$$Fitness\left(x(i)\right) = Maximum\_accuracy\left(x(i)\right) \quad (12)$$

Where, $x(i)$ is a random feature subset and $Maximum\_accuracy\left(x(i)\right)$ is an accuracy of random feature subset. The maximum number of iterations is 100 and fitness is calculated for every iteration and best fitness is selected.

3.4 Classification

The unidirectional LSTM keeps and recollects contextual past data and it removes contextual future data. However, the BiLSTM addresses the issue through using dual independent LSTM such as forward and backwards. The BiLSTM captures intricate and temporal patterns present in the selected features which leads to informative presentation when compared to LSTM and RNN. It has an ability of bidirectional processing thereby capturing contextual data in LULC for differentiating classes. The BiLSTM has an exploding gradient issue during training which is solved through using the PFTS activation function. The forward LSTM processes the data in a forward direction which means from past to future. Whereas backward LSTM processes the data in reverse directions which means from future to past. The LSTM contains forget gate (f_t), input gate (i_t), output gate (o_t) and cell memory state (c_t). The input features $x = \{x_1, x_2, x_3, \dots, x_t\}$, the hidden state is stated in Eqs. (13)-(17).

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \quad (13)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (14)$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (15)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot PFTS(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (16)$$

$$h_t = o_t \odot PFTS(c_t) \quad (17)$$

Where, σ is sigmoid function. W_f, W_i and W_o are weight matrix, b_f, b_i and b_o are bias matrix of forget, input and output gates. The x_t is an LSTM input, h_t is a hidden layer vector. The major aim for using BiLSTM is a temporal pattern of scenes which are utilized among time series images. The input sequence is decided by BiLSTM that is stated as $i = i_1, i_2, \dots, i_n$ from opposite to forward order $\vec{f}_t = \left(\vec{f}_1, \vec{f}_2, \dots, \vec{f}_n\right)$ and backwards hidden order $\overleftarrow{f}_t = \left(\overleftarrow{f}_1, \overleftarrow{f}_2, \dots, \overleftarrow{f}_n\right)$. The v_t is an encoder vector that is estimated by gathering decision forward and backward output $v_t = \left[\vec{f}_t, \overleftarrow{f}_t\right]$ as Eqs. (18)-(20).

$$\vec{f}_t = \delta\left(W_{\vec{f}_i} i_t + W_{\vec{f}\vec{f}} \vec{f}_{t-1} + q_{\vec{f}}\right) \quad (18)$$

$$\overleftarrow{f}_t = \delta\left(W_{\overleftarrow{f}_i} i_t + W_{\overleftarrow{f}\overleftarrow{f}} \overleftarrow{f}_{t-1} + q_{\overleftarrow{f}}\right) \quad (19)$$

$$v_t = W_{v_{\vec{f}}} \vec{f}_t + W_{v_{\overleftarrow{f}}} \overleftarrow{f}_t + q_v \quad (20)$$

Where, the δ is a logistic sigmoid function, $v = v_1, v_2, \dots, v_t, \dots, v_n$ is an initial hidden layer output sequence.

3.4.1. Parametric Flatten-T Swish

The FTS is a predefined activation function and it does not produce advantages such as non-linear representation, flexibility and dynamicity ability for networks. Therefore, PFTS is used to overcome this limitation. The PFTS learns entire activation function and hinge point t_{train} of function which enables to vertically fluctuate and determine optimal value for t_{train}. The PFTS is stated in Eq. (21).

$$PFTS(x) = \begin{cases} \frac{x}{1+e^{-x}} + t_{train}, & if\ x \geq 0 \\ t_{train}, & if\ x < 0 \end{cases} \quad (21)$$

Where, x is an activation function input, t_{train} is an adaptive parameter that enables learning from training.

4. Result analysis

The proposed PWMA-BiLSTM with PFTS is simulated in MATLAB of 2020a version with system configuration of i5 processor, windows 10 OS and 16GB RAM. The PWMA-BiLSTM with PFTS is evaluated with the measure of accuracy, sensitivity, precision, f1-score, specificity and computational time. The TP, FP, TN and FN are true positive, false positive, true negative and false negative. Table 1 shows the metrics with their mathematical expressions.

Table 1. Metrics with its mathematical expression

Metrics	Mathematical Expression
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}\times 100$
Sensitivity	$\frac{TP}{TP+FN}\times 100$
Precision	$\frac{TP}{TP+FP}\times 100$
F1-Score	$\frac{2\times Precision\times Sensitivity}{Precision+Sensitivity}\times 100$
Specificity	$\frac{TN}{TN+FP}\times 100$

Table 2 presents the PWMA results for different feature selections with measures of accuracy, sensitivity, precision, f1-score and specificity. The different feature selection methods such as Flower Pollination Algorithm (FPA), Moth Search Algorithm (MSA) and WMA are used to evaluate the PWMA performance. The PWMA attains 99.95%, 99.82%, 98.68%, 99.24% and 99.91% of accuracy, sensitivity, precision, f1-score and specificity for the AID dataset. Additionally, PWMA attains 99.87%, 99.72%, 98.51%, 99.11% and 99.83% of accuracy, sensitivity, precision, f1-score and specificity for UCM dataset.

In Table 3, the result of BiLSTM with PFTS for different classification (actual features) are presented along with measures of accuracy, sensitivity, precision, f1-score and specificity. The different classification methods such as RNN, LSTM and BiLSTM are used to evaluate the BiLSTM with PFTS performance. The BiLSTM with PFTS attains 97.81%, 97.56%, 96.28%, 96.91% and 97.79% of accuracy, sensitivity, precision, f1-score and specificity for AID dataset. Additionally, BiLSTM with PFTS attains 97.53%, 97.47%, 96.31%, 96.88% and 97.46% of accuracy, sensitivity, precision, f1-score and specificity for UCM dataset.

Table 2. Different feature selection result

Dataset	Performance Metrics	FPA	MSA	WMA	PWMA
AID	Accuracy (%)	93.76	94.51	97.82	99.95
	Sensitivity (%)	93.55	94.38	97.21	99.82
	Precision (%)	92.37	93.83	96.46	98.68
	F1-Score (%)	92.95	94.10	96.83	99.24
	Specificity (%)	93.62	94.47	97.75	99.91
UCM	Accuracy (%)	93.54	94.26	97.58	99.87
	Sensitivity (%)	93.16	93.85	97.37	99.72
	Precision (%)	92.82	93.71	96.15	98.51
	F1-Score (%)	92.98	93.77	96.75	99.11
	Specificity (%)	93.47	94.37	97.61	99.83

Table 3. Classification result with actual features

Dataset	Performance Metrics	RNN	LSTM	BiLSTM	BiLSTM with PFTS
AID	Accuracy (%)	92.59	93.76	95.64	97.81
	Sensitivity (%)	92.43	93.52	95.31	97.56
	Precision (%)	91.37	92.34	94.73	96.28
	F1-Score (%)	91.89	92.92	95.01	96.91
	Specificity (%)	92.51	93.60	95.46	97.79
UCM	Accuracy (%)	92.57	93.65	95.52	97.53
	Sensitivity (%)	92.40	93.43	95.14	97.47
	Precision (%)	91.82	92.25	94.69	96.31
	F1-Score (%)	92.10	92.83	94.91	96.88
	Specificity (%)	92.48	93.57	95.37	97.46

Table 4. Classification result with optimized features

Dataset	Performance Metrics	RNN	LSTM	BiLSTM	BiLSTM with PFTS
AID	Accuracy (%)	93.82	95.67	96.73	99.95
	Sensitivity (%)	93.46	95.29	96.38	99.82
	Precision (%)	92.39	94.37	95.19	98.68
	F1-Score (%)	92.92	94.82	95.78	99.24
	Specificity (%)	93.71	95.59	96.62	99.91
UCM	Accuracy (%)	93.62	95.48	96.51	99.87
	Sensitivity (%)	93.35	95.24	96.42	99.72
	Precision (%)	92.40	94.67	95.27	98.51
	F1-Score (%)	92.87	94.95	95.84	99.11
	Specificity (%)	93.19	95.36	96.48	99.83

In Table 4, the result of BiLSTM with PFTS for different classification (optimized features) are presented along with measures of accuracy, sensitivity, precision, f1-score and specificity. The different classification methods such as RNN, LSTM and BiLSTM are used to evaluate the BiLSTM with PFTS performance. The BiLSTM with PFTS attains 99.95%, 99.82%, 98.68%, 99.24% and 99.91% of accuracy, sensitivity, precision, f1-score and specificity for AID dataset. Additionally, BiLSTM with PFTS attains 99.87%, 99.72%, 98.51%, 99.11% and 99.83% of accuracy, sensitivity, precision, f1-score and specificity for UCM dataset.

In Table 5, the results of BiLSTM with PFTS for different classifications are presented along with a measure of computation time for both AID and UCM datasets. The different classification methods such as RNN, LSTM and BiLSTM are used to evaluate the BiLSTM with PFTS performance. The BiLSTM with PFTS attains less computation time of 047sec, 052sec for AID and UCM datasets when compared to RNN, LSTM and BiLSTM. The BiLSTM with PFTS achieves less computation time because of effective gradient flow. Moreover, it has capability to process sequence in both forward and backward directions which quicken the convergence and reduced the overall computation time.

The confusion matrix for AID, UCM, RESISC45 and EcoCropsAID datasets are given in figure 7, 8, 9 and 10 respectively.

4.1 Comparative analysis

The proposed PWMA-BiLSTM with PFTS is compared with existing techniques such as DTLF-

Table 5. Computation time (sec) result

Dataset	AID	UCM
RNN	073	076
LSTM	061	064
BiLSTM	055	059
BiLSTM with PFTS	**047**	**052**

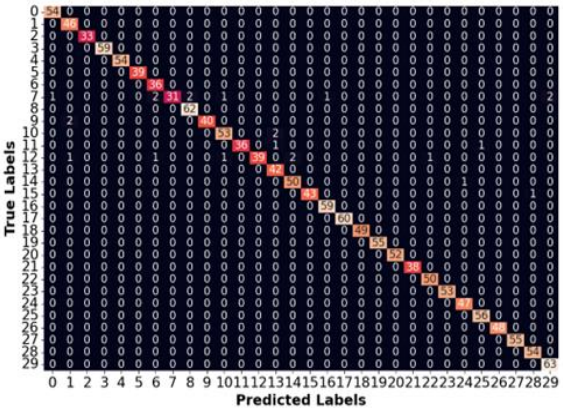

Figure. 7 Confusion matrix for AID dataset

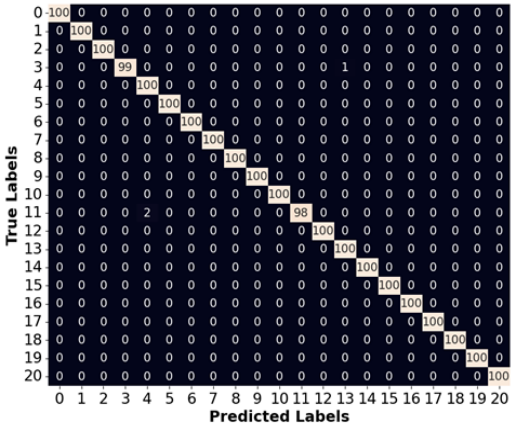

Figure. 8 Confusion matrix for UCM dataset

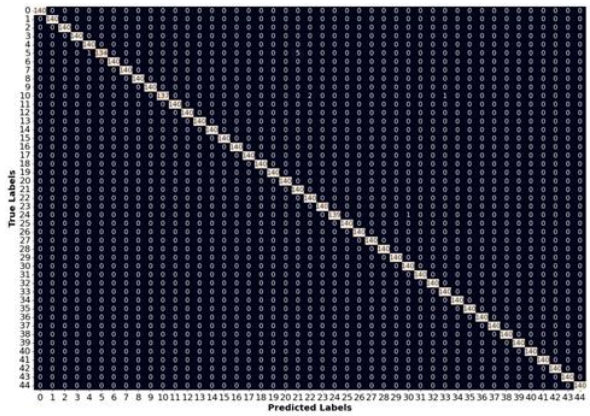

Figure. 9 Confusion matrix for RESISC45 dataset

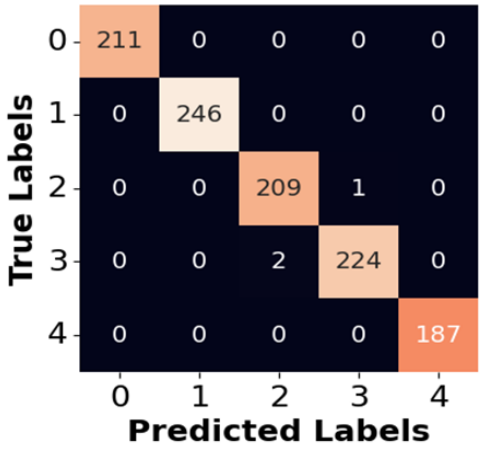

Figure. 10 Confusion matrix for EcoCropsAID dataset

ERSIC [17], FCN [18], FCMBS-RSIC [19], RS-CLIP [20], DropCyclic [21] and XE-Net [22] for AID, UCM, RESISC45 and EcoCropsAID datasets. The PWMA-BiLSTM with PFTS shows better performance than existing techniques in terms of accuracy, sensitivity, precision, f1-score, specificity and computation time. The PWMA-BiLSTM with PFTS attains 99.95%, 99.82%, 98.68%, 99.24%, 99.91% and 047sec of accuracy, sensitivity, precision, f1-score, specificity and computation time for the AID dataset as presented in table 6. Additionally, PWMA-BiLSTM with PFTS attains 99.87%, 99.72%, 98.51%, 99.11%, 99.83% and 052sec of accuracy, sensitivity, precision, f1-score, specificity and computation time for UCM dataset as presented in table 7. The PWMA-BiLSTM with PFMS attains 98.39%, 96.62%, 97.54% and 97.07% of accuracy, sensitivity, precision and f1-score for RESISC45 dataset as presented in table 8. The PWMA-BiLSTM with PFMS attains 99.76% of accuracy for EcoCropsAID dataset as presented in table 9.

4.2 Discussion

The existing techniques have limitations such as the DTLF-ERSIC [17] suffers from less classification accuracy due to gradient issues. FCN [18] produces higher dimensional features which reduces classification performance. FCMBS-RSIC [19] has

Table 6. Comparison of PWMA-BiLSTM with PFTS for AID dataset

Performance Metrics	**Methods**						
	DTLF-ERSIC [17]	**FCN [18]**	**FCMBS-RSIC [19]**	**RS-CLIP [20]**	**DropCyclic [21]**	**XE-Net [22]**	**PWMA-BiLSTM with PFTS**
Accuracy (%)	99.8	97.73	99.31	87.52	94.58	95.78	99.95
Sensitivity (%)	97.5	91.4	99.42	NA	NA	NA	99.82
Precision (%)	97.6	92.1	98.36	NA	NA	NA	98.68
F1-Score (%)	97.5	91.6	NA	NA	NA	NA	99.24
Specificity (%)	99.9	NA	NA	NA	NA	NA	99.91
Computation time (sec)	NA	NA	058	NA	NA	NA	047

Table 7. Comparison of PWMA-BiLSTM with PFTS for UCM dataset

Performance Metrics	**Methods**						
	DTLF-ERSIC [17]	**FCN [18]**	**FCMBS-RSIC [19]**	**RS-CLIP [20]**	**DropCyclic [21]**	**XE-Net [22]**	**PWMA-BiLSTM with PFTS**
Accuracy (%)	99.7	98.75	99.63	95.94	97.38	99.37	99.87
Sensitivity (%)	96.7	91.3	99.67	NA	NA	NA	99.72
Precision (%)	96.8	91.9	98.12	NA	NA	NA	98.51
F1-Score (%)	96.7	91.6	NA	NA	NA	NA	99.11
Specificity (%)	99.8	NA	NA	NA	NA	NA	99.83
Computation time (sec)	NA	NA	064	NA	NA	NA	052

Table 8. Comparison of PWMA-BiLSTM with PFTS for RESISC45 dataset

Performance Metrics	**Methods**			
	FCN [18]	**RS-CLIP [20]**	**XE-Net [22]**	**PWMA-BiLSTM with PFTS**
Accuracy (%)	96.57	85.76	95.03	98.39
Sensitivity (%)	93.9	NA	NA	96.62
Precision (%)	94.7	NA	NA	97.54
F1-Score (%)	94.2	NA	NA	97.07

Table 9. Comparison of PWMA-BiLSTM with PFTS for EcoCropsAID dataset

Performance Metrics	**Methods**	
	DropCyclic [21]	**PWMA-BiLSTM with PFTS**
Accuracy (%)	99.12	99.76

an inability for feature fusion due to high correlated features. RS-CLIP [20] has less convergence speed which leads less detection performance because it unable to learn optimal solutions. DropCyclic [21] has overfitting issue due to the lack of feature selection thereby reducing less classification performance. To overcome these issues, this research proposes a PWMA-BiLSTM with PFTS for LULC classification. The PWMA can search patterns and robust to noisy and inappropriate features thereby reducing optimal feature selection. The BiLSTM captures temporal and intricate patterns in selected features thereby leading informative presentation. By using BiLSTM with PFTS enhances the non-linear, smooth transformation by tunable parameters thereby leading better generalization and quick convergence in LULC classification.

5. Conclusion

The PWMA-BiLSTM with PFTS is proposed in this research for LULC classification. The PWMA enhances the capability to discover optimal features by retaining valuable information from past searches, quickening the convergence speed and reducing redundancy thereby leading to effective classification. It integrates global search abilities with personal local search memory for enhancing the exploration of feature subsets. The BiLSTM captures temporal and complex patterns and it has the ability of bidirectional processing to capture contextual data in LULC for distinguishing various classes. The gradient, texture and hierarchical features are extracted by HOG, LTP and CNN based feature extraction techniques from AID and UCM datasets. The PWMA-BILSTM with PFTS attains the accuracy of 99.95% and 99.87% for AID and UCM datasets respectively. In the future, hybrid metaheuristic optimization will be applied for LULC classification to further enhance accuracy.

Notation list

Notation	**Description**
s	Texture operator
t	User threshold
x, W and y	Residual block input, weight and output
P_S	Sound wave energy at sound source
r	Euclidean distance
x_i^t	Present position of woodpecker i in tth iteration
δ_i^t	Random vector of woodpecker i in tth iteration
r_1	Random number from normal distribution within a range of $[0, 1]$
x_{gpop}^t	Best member position of population
x_{mj}^t	jth male woodpecker
x_{ppop}^t	Memory location
α $(\alpha_{gpop}, \alpha_{mj})$	Parameter
r_3	Random number from normal distribution within a range of $[0, 3]$
$Tansig$	Tangent sigmoid function
t and t_{max}	Number of present and maximum iterations
α	Female woodpecker step size
SI_j^i	Sound intensity target
N	Woodpeckers' population size
$H\alpha$	Highest level of α
x_{RRA}^i	New element position
lb and ub	Lower and upper bound variables
r_4	Random number from normal distribution within a range of $[0, 1]$
x_{GRA}^i	New element location attained from GRA on ith woodpecker
x_i^t	Random woodpecker location

GRA_{bit}	Binary vector
x_{gpop}^{t}	Location of best woodpecker over the population
x_r	Random woodpecker location
R	Random numbers from random distribution within a range of $[-1, 1]$
$x(i)$	Random feature subset
$Maximum\_accurac$	Accuracy of random feature subset
f_t	Forget gate
i_t	Input gate
o_t	Output gate
c_t	Memory cell state
σ	Sigmoid function
W_f, W_i and W_o	Weight matrix of forget, input and output gates
b_f, b_i and b_o	Bias matrix of forget, input and output gates
x_t	LSTM input
h_t	Hidden layer vector
v_t	Encoder vector
δ	Logistic sigmoid function
x	Activation function input
t_{train}	Adaptive parameter
TP	True positive
TN	True negative
FP	False positive
FN	False negative

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 3rd, 5th and 4th author. The supervision and project administration, have been done by 1st and 2nd author.

References

[1] C. Pushpalatha, B. Sivasankari, A. Ahilan, and K. Kannan, "Landscape Classification Using an Optimized Ghost Network from Aerial Images", *Journal of the Indian Society of Remote Sensing*, 2024.

[2] A. Kumar, and A.K. Gorai, "Application of transfer learning of deep CNN model for classification of time-series satellite images to assess the long-term impacts of coal mining activities on land-use patterns", *Geocarto International*, Vol. 37, No. 26, pp. 11420-11440, 2022.

[3] W. Miao, J. Geng, and W. Jiang, "Multigranularity decoupling network with pseudolabel selection for remote sensing image scene classification", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 61, p. 5603813, 2023.

[4] N.A. Azeem, S. Sharma, and S. Hasija, "Classification of Satellite Images Using an Ensembling Approach Based on Deep Learning", *Arabian Journal for Science and Engineering*, Vol. 49, No. 3, pp. 3703-3718, 2024.

[5] M.S. Minu, and R.A. Canessane, "Deep learning-based aerial image classification model using inception with residual network and multilayer perceptron", *Microprocessors and Microsystems*, Vol. 95, p. 104652, 2022.

[6] P. Prema, and V.V. Ramalingam, "OptiLCD: an optimal lossless compression and denoising technique for satellite images using hybrid optimization and deep learning techniques", *Soft Computing*, Vol. 27, No. 24, pp. 18605-18622, 2023.

[7] O. Akar, and E.T. Gormus, "Land use/land cover mapping from airborne hyperspectral images with machine learning algorithms and contextual information", *Geocarto International*, Vol. 37, No. 14, pp. 3963-3990, 2022.

[8] S.S. Virnodkar, V.K. Pachghare, V.C. Patil, and S.K. Jha, "CaneSat dataset to leverage convolutional neural networks for sugarcane classification from Sentinel-2", *Journal of King Saud University-Computer and Information Sciences*, Vol. 34, No. 6B, pp. 3343-3355, 2022.

[9] S.D. Thepade, and S. Chauhan, "Enhancing Land Use Identification Through Fusion of Densenet121 DCNN and Thepade SBTC Features Using Machine Learning Algorithms and Ensembles", *SN Computer Science*, Vol. 4, p. 772, 2023.

[10] X. Li, C. Sun, H. Meng, X. Ma, G. Huang, and X. Xu, "A novel efficient method for land cover classification in fragmented agricultural landscapes using sentinel satellite imagery", *Remote Sensing*, Vol. 14, No. 9, p. 2045, 2022.

[11] M. Stoimchev, J. Levatić, D. Kocev, and S. Džeroski, "Semi-Supervised Multi-Label Classification of Land Use/Land Cover in Remote Sensing Images with Predictive Clustering Trees and Ensembles", *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 62, p. 4706416, 2024.

[12] A.U. Siddiqui, and M.K. Jain, "Change analysis in land use land cover due to surface mining in Jharia coalfield through Landsat time series data", *Materials Today: Proceedings*, Vol. 49, Part 8, pp. 3462-3468, 2022.

[13] I. Papoutsis, N.I. Bountos, A. Zavras, D. Michail, and C. Tryfonopoulos, "Benchmarking and scaling of deep learning models for land cover image classification", *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 195, pp. 250-268, 2023.

[14] T. Qiu, H. He, X. Liang, F. Chen, Z. Chen, and Y. Liu, "Using different training strategies for urban land-use classification based on convolutional neural networks", *Frontiers in Environmental Science*, Vol. 10, p. 981486, 2022.

[15] X. Yu, K. Zhang, and Y. Zhang, "Land use classification of open-pit mine based on multi-scale segmentation and random forest model", *Plos One*, Vol. 17, No. 2, p. e0263870, 2022.

[16] D. Lin, and Z. Chen, "Semantic understandings for aerial images via multigrained feature grouping", *Scientific Programming*, Vol. 2022, No. 1, p. 1822539, 2022.

[17] A.M. Hilal, F.N. Al-Wesabi, K.J. Alzahrani, M. Al Duhayyim, M.A. Hamza, M. Rizwanullah, and V.G. Díaz, "Deep transfer learning based fusion model for environmental remote sensing image classification model", *European Journal of Remote Sensing*, Vol. 55, Supp. 1, pp. 12-23, 2022.

[18] Y.Y. Ghadi, A.A. Rafique, T. Al Shloul, S.A. Alsuhibany, A. Jalal, and J. Park, "Robust object categorization and Scene classification over remote sensing images via features fusion and fully convolutional network", *Remote Sensing*, Vol. 14, No. 7, p. 1550, 2022.

[19] A.M. Hilal, H. Alsolai, F.N. Al-Wesabi, M.K. Nour, A. Motwakel, A. Kumar, I. Yaseen, and A.S. Zamani, "Fuzzy Cognitive Maps with Bird Swarm Intelligence Optimization-Based Remote Sensing Image Classification", *Computational Intelligence and Neuroscience*, Vol. 2022, No. 1, p. 4063354, 2022.

[20] X. Li, C. Wen, Y. Hu, and N. Zhou, "RS-CLIP: Zero shot remote sensing scene classification via contrastive vision-language supervision", *International Journal of Applied Earth Observation and Geoinformation*, Vol. 124, p. 103497, 2023.

[21] S. Noppitak, and O. Surinta, "dropCyclic: snapshot ensemble convolutional neural network based on a new learning rate schedule for land use classification", *IEEE Access*, Vol. 10, pp. 60725-60737, 2022.

[22] N. Guo, M. Jiang, D. Wang, X. Zhou, Z. Song, Y. Li, L. Gao, and J. Luo, "Scene classification for remote sensing image of land use and land cover using dual-model architecture with multilevel feature fusion", *International Journal of Digital Earth*, Vol. 17, No. 1, p.2353166, 2024.

[23] AID dataset link: https://paperswithcode.com/dataset/aid (accessed on August 2024).

[24] UCM dataset link: http://weegee.vision.ucmerced.edu/datasets/landuse.html (accessed on August 2024).

[25] K.S. Neetha, and D.L. Narayan, "Feature selection using adaptive manta ray foraging optimization for brain tumor classification", *Pattern Analysis and Applications*, Vol. 27, p. 29, 2024.

[26] B.B. Rudra, and G. Murtugudde, "Hybrid Feature Selection with Parallel Multi-Class Support Vector Machine for Land Use Classification", *International Journal of Intelligent Engineering & Systems*, Vol. 15, No. 1, pp. 85-94, 2022, doi: 10.22266/ijies2022.0228.09.