INNOVATIVE MATHEMATICAL MODELLING APPROACHES TO DIAGNOSE CHRONIC NEUROLOGICAL DISORDERS WITH DEEP LEARNING

by

Faten K. KARIM [a]***, Sara GHORASHI*** [a*]***, Anis Ben ISHAK*** [b]***,***
Azhari A. ELHAG [c]***, and Nahla MOHAMED*** [d]

[a] Department of Computer Sciences, College of Computer and Information Sciences,
Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia
[b] Department of Quantitative Methods, Higher Institute of Management, University of Tunis, Tunis, Tunisia
[c] Department of Mathematics and Statistics, College of Science, Taif University, Taif, Saudi Arabia
[d] Uppsala Universitet, Uppsala, Sweden

Original scientific paper
https://doi.org/10.2298/TSCI2406217K

Multiple sclerosis impacts the central nervous system, causing symptoms like fatigue, pain, and motor impairments. Diagnosing multiple sclerosis often requires complex tests, and MRI analysis is critical for accuracy. Machine learning has emerged as a key tool in neurological disease diagnosis. This paper introduces the multiple sclerosis diagnosis network (MSDNet), a stacked ensemble of deep learning classifiers for multiple sclerosis detection. The MSDNet uses min-max normalization, the artificial hummingbird algorithm for feature selection, and a combination of LSTM, DNN, and CNN models. Hyperparameters are optimized using the enhanced walrus optimization algorithm. Experimental results show MSDNet's superior performance compared to recent methods.

Key words: *chronic neurodegenerative disorder, advanced machine learning, refined walrus optimization method, bioinspired optimization technique*

Introduction

Multiple sclerosis (MS) is a chronic disorder affecting the central nervous system (CNS) through neurodegeneration and inflammation, leading to symptoms like blurred vision, muscle weakness, and fatigue [1]. It is characterized by the formation of lesions or plaques in the white matter of the CNS, which disrupt neural communication [2]. The MS is classified into types such as relapsing-remitting MS (RRMS) and primary progressive MS (PPMS), with RRMS being the most common, involving periods of relapse followed by remission [3]. Diagnosing MS remains a complex process, often involving MRI scans to detect brain lesions, however, manual lesion identification can be time-consuming and prone to human error [4, 5]. As a result, there is an increasing interest in automating MS diagnosis using AI, particularly through machine learning (ML) and deep learning (DL) techniques, which can efficiently analyze medical imaging data and reduce diagnostic errors [6-10].

Recent advancements have demonstrated the potential of AI in enhancing MS diagnosis. Yaghoubi *et al.* [11] introduced a novel method using scanning laser ophthalmoscopy images to detect retinal vessel abnormalities, offering a non-MRI-based approach to identifying

[*] Corresponding author, e-mail: saabdelghani@pnu.edu.sa

vascular biomarkers associated with MS. Mohammed *et al.* [12] leveraged DL, particularly convolutional neural networks (CNN), to improve the speed and accuracy of MS diagnosis through MRI data analysis. Their work highlighted CNNs' ability to capture complex patterns in medical images, which is essential for early detection of MS lesions. Langat *et al.* [13] advanced this concept further by integrating 6G-enabled Internet of Medical Things technology with DenseNets, showing significant improvements in diagnostic accuracy, particularly in cases of acute transverse myelitis. Langat *et al.* [14] focused on refining lesion segmentation using the MSAT algorithm and HDCARAN classifier, demonstrating enhanced precision in MRI analysis, while Ponce de Leon-Sanchez *et al.* [15] explored various DL architectures, including LSTM-CNN and GAN, to understand their applications across different neurological diseases. Moreover, Al Jannat *et al.* [16] employed an ANN model to predict MS diagnosis, achieving high accuracy by selecting relevant features from gene expression profiles. This underscores the growing role of genetic data in augmenting neuroimaging techniques. Rode and Soddamallaiah [17] developed a CNN-based system that utilized transfer learning and softMax for classifying MS progression, successfully dealing with the challenge of sparse lesion data. Finally, Kappal [18] proposed the PAFEM-IS feature extraction method to optimize MS lesion segmentation, further demonstrating the power of AI-based solutions in minimizing diagnostic errors and reducing time consumption in clinical settings. These approaches reflect the ongoing shift towards automated and more reliable MS diagnosis through AI technologies.

This study builds on these advancements by proposing the MSDNet, a framework that employs min-max normalization standardize data, the artificial hummingbird algorithm (AHA) for feature selection, and a stacked ensemble of DL models (LSTM, DNN, CNN) for MS diagnosis. The integration of multiple DL models in a stacked configuration aims to maximize diagnostic accuracy by capturing diverse data features. Hyperparameter tuning is further optimized using the enhanced walrus optimization algorithm (EWOA), which fine-tunes the performance of the ensemble models. Experimental results indicate that the MSDNet framework outperforms recent approaches, offering a more robust and accurate method for MS diagnosis under various testing conditions.

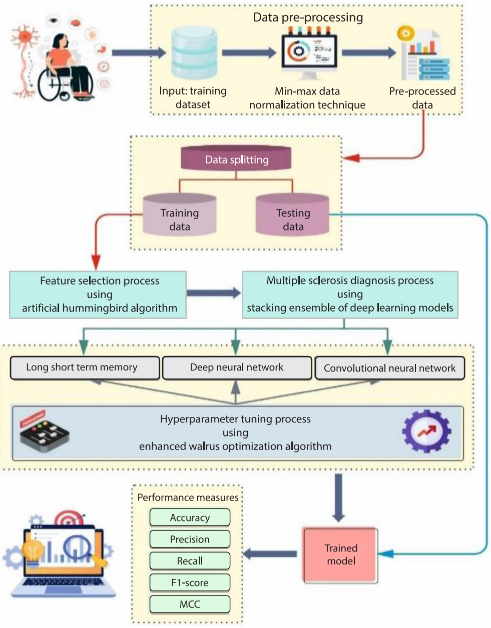
Data pre-processing
Input: training dataset
Min-max data normalization technique
Pre-processed data
Data splitting
Training data
Testing data
Feature selection process using artificial hummingbird algorithm
Multiple sclerosis diagnosis process using stacking ensemble of deep learning models
Long short term memory
Deep neural network
Convolutional neural network
Hyperparameter tuning process using enhanced walrus optimization algorithm
Performance measures
Accuracy
Precision
Recall
F1-score
MCC
Trained model

Figure 1. The MSDNet architecture flow

Proposed methodology

This study presents the MSDNet model for diagnosing MS, which involves data normalization, feature selection, a stacked ensemble of DL classifiers, and hyperparameter tuning. The workflow is depicted in fig. 1.

Feature Selection Process

The MSDNet technique starts with min-max normalization, standardizing data to a range of 0 to 1, which is vital in MS analysis [19]. This ensures all features, like MRI readings and clinical scores, are treated equally, preventing bias. This normalization enhances ML's ability to differentiate between healthy individuals and those with MS, improving diagnostic accuracy and effectively managing diverse data types in MS diagnosis. Additionally, the feature selection process uses the AHA to identify the optimal subset of features. This algorithm mimics a hummingbird's behavior of exploring various food sources, with each *hummingbird* retaining memories of encountered sources, and consolidating data from multiple sources to determine the best features [20].

Initialize

The process begins by placing *mmm* hummingbirds on *mmm* food sources:

$$c_r = Low + i\left(Up - Low\right), \quad r = 1,\dots,m \tag{1}$$

where the d-dimensional upper and lower bounds are denoted by Up and Low. A randomly generated vector in the interval [0, 1] was designated by i, and the provided issue's solution r^{th} food resource's position is represented by c_r.

Guided foraging: Hummingbirds possess the remarkable ability to locate food sources that contain substantial amounts of liquid. In the AHA methodology, three key abilities – axial, omnidirectional, and sloping – are utilized during the foraging process, with a route-switching trajectory designed for effective navigation.

The axial fight in the w – W space was provided:

$$W^{(r)} = \begin{cases} 1 \text{ if } r = randi\left(\left[1,\ w\right]\right) \\ 0 \text{ else} \end{cases} \quad r = 1,\ \dots,\ w \tag{2}$$

The diagonal fight has been determined:

$$W^{(r)} = \begin{cases} 1 \text{ if } r = k\left(q\right),\ q \in \left[1, p\right], \\ k = randperm\left(p\right), \\ p \in \left[2,\ i_1 \cdot \left(w-2\right)+1\right] \\ 0 \text{ else} \end{cases} \quad i = 1,\ \dots,\ w \tag{3}$$

The explanation of the omnidirectional fight is assumed:

$$W^{(r)} = 1, \quad r = 1,\ \dots,\ w \tag{4}$$

where from 1 to w a random number is created by $randi([1, w])$ from 1 to p an uneven transformation of numbers is produced by $randperm(p)$, and a randomly produced value in the range of [0, 1] is i_1. The numerical calculation of foraging is shown:

$$E_r = \left(g+1\right) = c_{r,tar}\left(S\right) + zW\left(c_r\left(g\right) - c_{r,tar}\left(g\right)\right) \tag{5}$$

$$z \sim M\left(0,1\right) \tag{6}$$

At time g, the position of r^{th} food resource is denoted by $c(g)$, the portion of the r^{th} hummingbird reflects to stay was specified by $c_{r,tar}$(g), and the guided factor that experiences normal distribution $M(0, 1)$ is a. The r^{th} food resource's position upgrade is expressed:

$$c_r\left(g+1\right) = \begin{cases} c_r\left(+\right) u\left(c_r\left(g\right)\right) \le u\left(e_r\left(g+1\right)\right) \\ e_r\left(g+1\right) u\left(c_r\left(g\right)\right) > u\left(e_r\left(g+1\right)\right) \end{cases} \tag{7}$$

where the value of function fitness is indicated by u.

Territorial foraging: When a depleted food source is completely consumed, the hummingbird will seek a new source within its vicinity. The mathematical formulation for this local search process:

$$e_r\left(g+1\right)=c_r\left(g\right)+yW_C\left(g\right) \tag{8}$$

$$y\sim M\left(0{,}1\right) \tag{9}$$

In this context, the territorial factor follows a standard normal distribution $M(0,1)$ $M(0,1)M(0,1)$ represented as *yyy*. The movement from one location another, particularly towards the area with the least nectar replenishment, can be expressed:

$$C_{\text{worst}}\left(S+1\right)=LOW+i\left(UP-LOW\right) \tag{10}$$

where C_{worst} is the supply of food with the poorest nectar replacement rate in the population. Assuming a 50% likelihood for both regional and guided foraging strategies, the guided foraging method has an equal probability of remaining at various resources. At this point, the migration strategy needs to be implemented to enhance efficiency and identify suitable hunting grounds:

$$N=2m \tag{11}$$

The computational intricacy is related to initialization, the health assessment (x_{eval}), the hummingbird size of population (N_{size}), the most iterations count ($T_{\max}$), and the measurement of variables (d_{var}). Then:

$$\begin{aligned}&O\left(AHO\right)=O\left(\text{problem definition}\right)+O\left(\text{initialization}\right)+O\left(S\left(\text{evaluationunction}\right)\right)+\\&+O\left(g\left(\textit{guided foraging}\right)\right)+O\left(g\left(\textit{teritorial foraging}\right)\right)+O\left(g\left(\textit{migration foraging}\right)\right)=\\&=O\left(1+mw+Gxm+\frac{1}{2}Gmw+\frac{1}{2}Gmw\frac{G}{2m}mx\right)\cong O\left(Gxm+Gmw+\frac{Gw}{2}\right)\end{aligned} \tag{12}$$

Searching features: Dual test functions are employed to demonstrate the search features of AHA. RosenBrock is a popular test problem for gradient-based optimization algorithms, it is a unimodal function is the first function. The $c = (1, 1)$ with $u(c) = 0$ is the optimum solution for this purpose. The *Rastrigin* function is the 2nd function and $c = (O, 0)$ with $u(c) = 0$ is its optimal solution.

In the AHA, multiple objectives are merged into a single objective function, with each objective's importance represented by a specific weight. In this study, we introduce a fitness function that integrates both feature selection objectives, as illustrated:

$$\text{Fitness}\left(X\right)=\alpha E\left(X\right)+\beta^*\left(1-\frac{|R|}{|N|}\right) \tag{13}$$

where Fitness(X) signifies the fitness value of a sub-set X, $E(X)$ epitomizes the classifier rate of error by utilizing the nominated feature in the X sub-set, $|R|$ and $|N|$ are the number of nominated and original features, correspondingly, and α and β are weights of the classifier error and the lessening proportion, $\alpha \in [0, 1]$ and $\beta = (1 - \alpha)$.

Stacking ensemble of deep learning

Stacking is an ensemble method that utilizes multiple base learners (BL) to classify input data [21]. The classifiers from these level-0 learners serve as input for a meta-learner or Level 1 model, allowing diverse learners to independently classify data and enhancing analysis of varied data types. The stacking ensemble method incorporates three techniques: LSTM, DNN, and CNN, creating a larger neural network by combining the outputs of these models. Each BL leverages different techniques for classification, ensuring heterogeneity in data analysis while employing similar loss activation functions.

Instead of using a distinct method as the meta-learner, we developed a fully connected (FC) neural network layer to merge forecasts from the BL. This meta-learner conducts an additional training stage to refine the final result, forming a unified model that integrates four independent sub-models, an extra FC layer, and a resultant layer for final predictions. By combining various identification techniques from the sub-models, the approach effectively captures diverse features in the data.

The LSTM classifier, a specialized type of recurrent neural network (RNN), is adept at learning long-term dependencies in data, addressing issues of exploding and vanishing gradients in conventional RNN [22]. The LSTM units employ three types of gates to control data flow: the forget gate removes irrelevant information, the input gate retains new data, and the output gate decides what data to output. This structure enables LSTM to maintain cell state over time, facilitating long-sequence data processing. Figure 2 defines the infrastructure of LSTM.

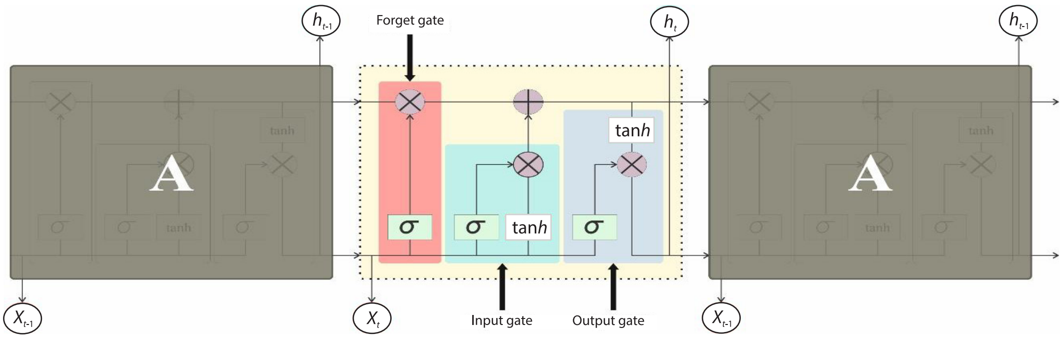

Figure 2. Architecture of LSTM

The DNN classifier, inspired by the architecture of the human brain, excels in pattern recognition and uncovering hidden correlations through a self-learning process [23]. The DNN consist of multiple layers of connected nodes, allowing them to model complex relationships in data. They utilize a backpropagation method to adjust weights and biases during training, enabling them to learn intricate patterns and make accurate predictions on new data.

The CNN, a specialized category of DL methods, utilize convolutional layers instead of traditional matrix multiplication, making them particularly effective for image data [24]. A CNN comprises feature extraction and classification components, where input data passes through convolutional and pooling layers to create feature maps. These maps are then flattened and processed by FC layers for classification. The CNN excel at capturing spatial relationships and patterns within data, optimizing feature extraction for complex datasets.

Fine-tuning the deep learning model

The EWOA-assisted hyperparameter tuning method enhances stacking DL models using the enhanced elephant and walrus optimization algorithm (EWOA), a nature-inspired metaheuristic inspired by walrus social behavior and sensory adaptations. Walruses, known for their advanced tactile senses and unique canine teeth, inform the EWOA's modelling of population behavior in response to safety alerts, leading to effective hyperparameter optimization. Key features of the EWOA approach are detailed in the subsequent sections:

- Initialization.
- The initial set of randomly generated candidate performances is used in the optimizer method within the variable bounds, with walrus location agents updated through iterations.

– Danger signals and safety signals:

The EWOA incorporates safety and danger alarms, which are crucial for understanding walrus behavior. The danger signal can be defined:

$$\text{Danger-signal} = A \times R \tag{14}$$

$$A = 2 \times \alpha \tag{15}$$

$$\alpha = 1 - \frac{t}{T} \tag{16}$$

$$R = 2 \times r_1 - 1 \tag{17}$$

$$\text{Safety-signal} = r_2 \tag{18}$$

where A and R sre the signify factor of danger, the factor α reduces from one to *zero* in the optimizer iterations. The random variables r_1 and r_2 are in *zero* and one. The safety signal can be described as r_2. The t is number of iterations, whereas T represents its most predefined number.

– *Migration (exploration):*

This phase signifies exploration in the method, the walrus location can be upgraded based on parameters containing a migration phase controlling factor β, and a randomly generated value r_3. The formula for upgrading the walrus location:

$$X_{ij}^{t+1} = X_{ij}^{t} + \text{Migration-step} \tag{19}$$

$$\text{Migration-step} = \left(X_m^t - X_n^t\right) \times \beta \times r_3^2 \tag{20}$$

$$\beta = 1 - \frac{1}{1 + \mathrm{e}^{\frac{-10(t-0.5T)}{T}}} \tag{21}$$

where X_{ij}^{t+1} is the novel location for the i^{th} iteration and j^{th} dimension, X_m^t and X_n^t are dual random chosen locations.

– *Reproduction (exploitation):*

In the reproduction phase, behaviors vary among juvenile, male, and female walruses. Male walrus positions are adjusted using the Halton sequence distribution, while female locations are influenced by either a male or the dominant walrus, as expressed:

$$\text{female}_{ij}^{t+1} = \text{female}_{ij}^{t} + \alpha \times \left(\text{male}_{ij}^{t} - \text{female}_{ij}^{t}\right) + (1-\alpha) \times \left(X_{\text{best}}^{t} - \text{female}_{ij}^{t}\right) \tag{22}$$

$$\text{Juvenile}_{ij}^{t+1} = \left(0 - \text{Juvenile}_{ij}^{t}\right) \times P \tag{23}$$

$$O = X_{\text{best}}^{t} + \text{Juvenile}_{ij}^{t} \times LF \tag{24}$$

where O is the reference safety position, PPP – the danger coefficient of young walruses, and the LF – the vector signifies Levy movement, enhancing optimization. The integration of Levy flight (LF) improves the original EWOA, enabling efficient exploration of large search spaces through long jumps and heavy tails. This adjustment enhances convergence, making the EWOA a more effective meta-heuristic optimizer for complex problems. The Levy function is defined:

$$LF = 0.01 \times \frac{u \times \sigma}{|v|^{1/\gamma}},\ \sigma = \left(\frac{\Gamma(1+\gamma) \times sin\left(\frac{\pi\gamma}{2}\right)}{\Gamma\left(\frac{1+\gamma}{2}\right) \times \gamma \times 2\left(\frac{\gamma-1}{2}\right)} 1^{1/\gamma} \tag{25}$$

where v, u is the stands for randomly generated value ranges from 0 to 1.

The fitness choice is crucial for the EEWOA system's solution. The parameter selection uses an encoding technique to evaluate candidate solutions, with accuracy prioritized in the EEWOA approach, expressed:

$$\text{Fitness} = \max\left(P\right) \tag{26}$$

$$P = \frac{TP}{TP + FP} \tag{27}$$

where TP implies the true positive and FP means the false positive rates.

Experimental results and analysis

This section validates the MSDNet technique using the MS disease dataset [25], which includes 273 groups across two classes as depicted in tab. 1. While the dataset has 19 features, 8 were selected: Gender, Age, Schooling, Breastfeeding, Varicella, Initial_Symptom, LLSSEP, and Periventricular_MRI.

Table 1. Details on dataset

Classes	Number of group
CDMS (clinical definite multiple sclerosis)	125
Non-CDMS (clinical definite multiple sclerosis)	148
Total number of group	273

Figure 3 displays the confusion matrices generated by the MSDNet model across different epochs, demonstrating effective recognition and classification of both classes. The MS disease recognition results of the MSDNet model, detailed in tab. 2, indicate that the system accurately identified every sample.

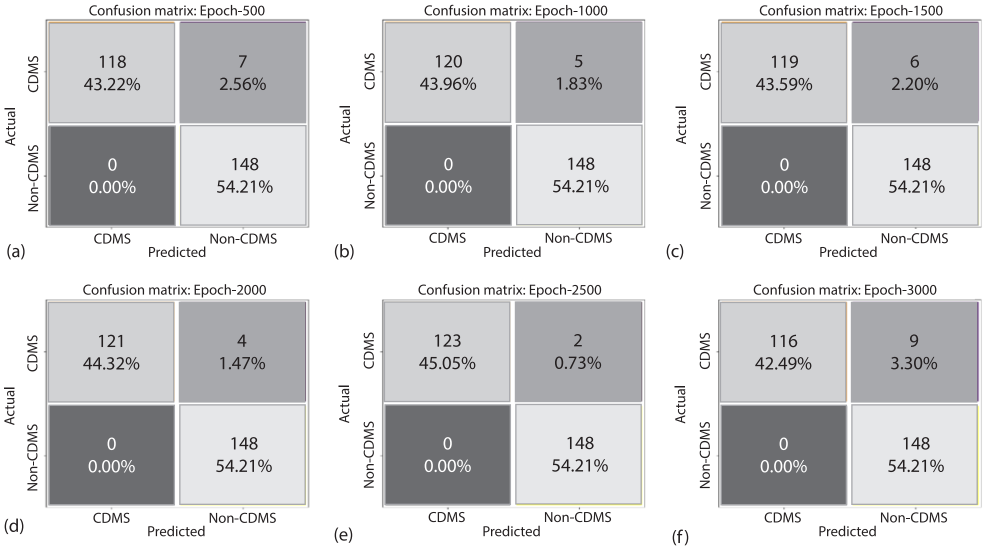

Figure 3. Confusion matrices of MSDNet technique; (a)-(f) epochs 500-3000

Table 2. The MS disease recognition outcome of MSDNet technique under distinct epochs

Class	$accu_y$	$prec_n$	$reca_l$	$F1_{\text{score}}$	MCC
Epoch – 500					
CDMS	97.44	100.00	94.40	97.12	94.94
Non-CDMS	97.44	95.48	100.00	97.69	94.94
Average	97.44	97.74	97.20	97.40	94.94
Epoch – 1000					
CDMS	98.17	100.00	96.00	97.96	96.37
Non-CDMS	98.17	96.73	100.00	98.34	96.37
Average	98.17	98.37	98.00	98.15	96.37
Epoch – 1500					
CDMS	97.80	100.00	95.20	97.54	95.65
Non-CDMS	97.80	96.10	100.00	98.01	95.65
Average	97.80	98.05	97.60	97.78	95.65
Epoch – 2000					
CDMS	96.80	100.00	96.80	98.37	97.08
Non-CDMS	100.00	97.37	100.00	98.67	97.08
Average	98.40	98.68	98.40	98.52	97.08
Epoch – 2500					
CDMS	99.27	100.00	98.40	99.19	98.53
Non-CDMS	99.27	98.67	100.00	99.33	98.53
Average	99.27	99.33	99.20	99.26	98.53
Epoch – 3000					
CDMS	92.80	100.00	92.80	96.27	93.53
Non-CDMS	100.00	94.27	100.00	97.05	93.53
Average	96.40	97.13	96.40	96.66	93.53

Figure 4 delivers the average result of the MSDNet system under epochs 500-1500. On 500 epochs, the MSDNet model gets an average $accu_y$ of 97.44%, $prec_n$ of 97.74%, $reca_l$ of 97.20%, $F1_{\text{score}}$ of 97.40%, and MCC of 94.94%. Moreover, on 1000 epochs, the MSDNet method provides an average $accu_y$ of 98.17%, $prec_n$ of 98.37%, $reca_l$ of 98.00%, $F1_{\text{score}}$ of 98.15%, and MCC of 96.37%. In the meantime, on 1500 epochs, the MSDNet technique offers an average $accu_y$ of 97.80%, $prec_n$ of 98.05%, $reca_l$ of 97.60%, $F1_{\text{score}}$ of 97.78%, and MCC of 95.65%.

Figure 5 offers an average result of the MSDNet approach under Epochs 2000- 3000. Based on 2000 epochs, the MSDNet methodology gets an average $accu_y$ of 98.40%, $prec_n$ of 98.68%, $reca_l$ of 98.40%, $F1_{\text{score}}$ of 98.52%, and MCC of 97.08%. Based on 2500 epochs, the MSDNet model provides an average $accu_y$ of 99.27%, $prec_n$ of 99.33%, $reca_l$ of 99.20%, $F1_{\text{score}}$ of 99.26%, and MCC of 98.53%. Based on 3000 epochs, the MSDNet technique delivers an average accuy of 96.40%, $prec_n$ of 97.13%, $reca_l$ of 96.40%, $F1_{\text{score}}$ of 96.66%, and MCC of 93.53%.

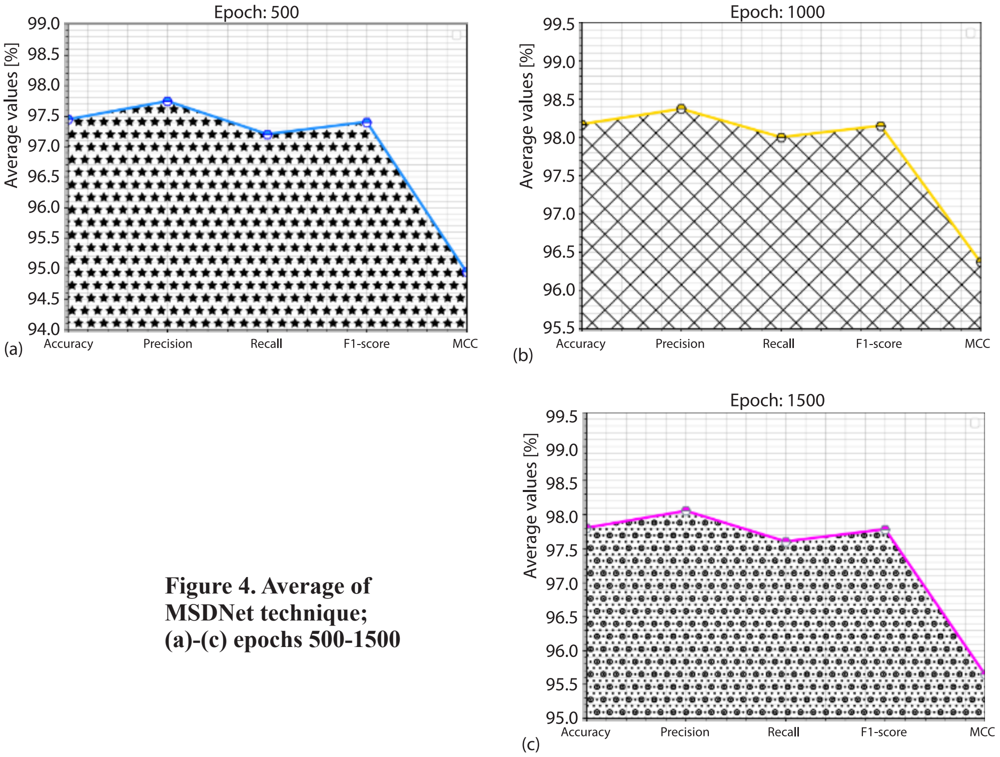

Figure 4. Average of MSDNet technique; (a)-(c) epochs 500-1500

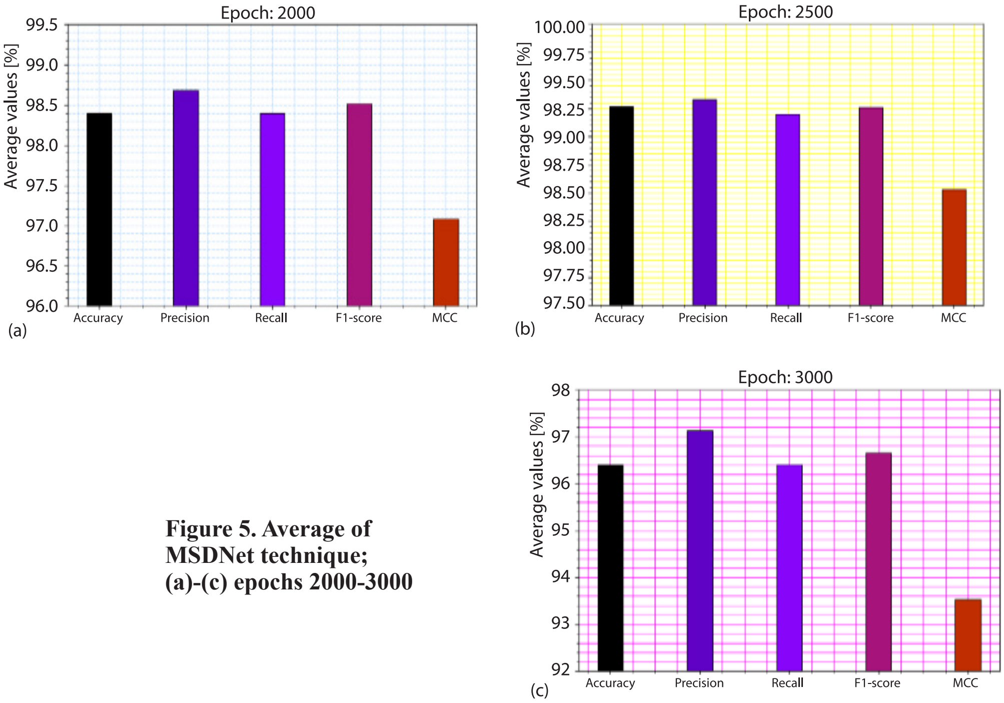

Figure 5. Average of MSDNet technique; (a)-(c) epochs 2000-3000

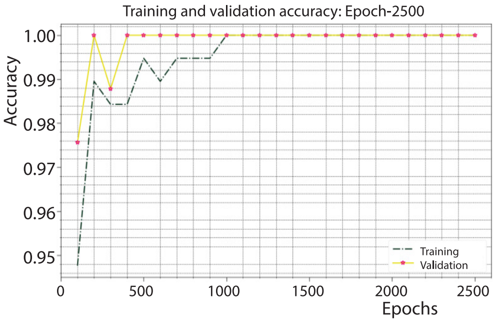

Figure 6. The $accu_y$ curve of MSDNet technique on epoch 2500

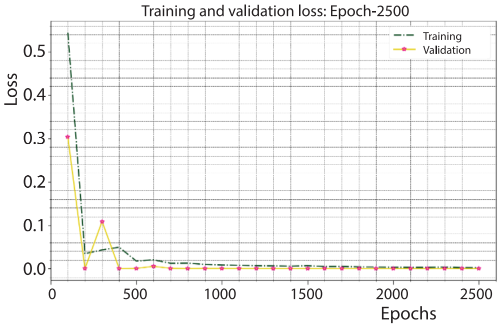

Figure 7. Loss curve of MSDNet technique on epoch 2500

Figure 6 illustrates the training (TRA) and validation (VLA) accuracy of the MSDNet technique over 2500 epochs. The accuracy values, spanning from 0 to 2500 epochs, show a consistent upward trend, indicating the model's improved performance with more iterations. Additionally, the close alignment of TRA and VLA accuracy suggests minimal overfitting, highlighting the reliability of the MSDNet technique for accurate predictions on unseen samples.

Figures 7-9 demonstrate the performance validation of the MSDNet system at 2500 epochs. Figure 7 shows a decreasing trend in TRA and VLA loss rates, indicating the model's effectiveness in balancing generalization and data fitting. Figure 8 presents the precision-recall (PR) curve, highlighting consistently high PR values across classes, which reflect the model's ability to capture true positives. Figure 9 displays the receiver operating characteristic (ROC) curve, showing high ROC scores for each class, underscoring the strong classification capability of the MSDNet approach.

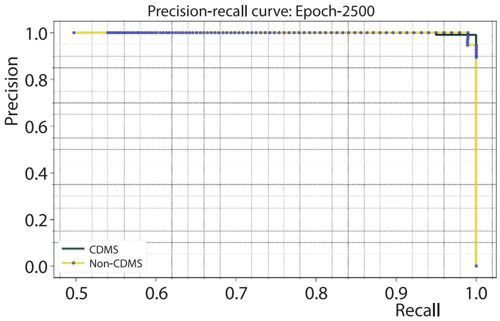

Figure 8. The PR curve of MSDNet technique on epoch 2500

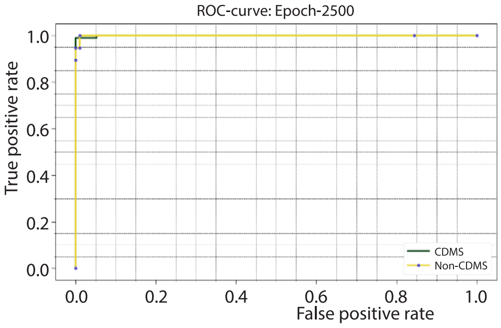

Figure 9. The ROC curve of MSDNet technique on epoch 2500

The comparative analysis of MSDNet methodology with recent techniques is demonstrated in tab. 3 [16, 26-28]. Figure 10 delivers comparison results of the MSDNet system with existing models in terms of $accu_y$. Based on $accu_y$, the MSDNet model has a greater $accu_y$ of 99.27% while the ANN, RF, GBM, RKPCA, SVM, NB, Ensemble Method and GNB methods have lesser $accu_y$ of 94.64%, 89.80%, 93.67%, 90.59%, 98.03%, 98.40%, 97.93%, and 98.65%, correspondingly.

Figure 11 provides comparison results of the MSDNet approach with recent techniques in terms of $prec_n$, $reca_l$, and $F1_{\text{score}}$. Depending upon $prec_n$ the MSDNet model has a greater $prec_n$, of 99.33% while the ANN, RF, GBM, RKPCA, SVM, NB, Ensemble Method and GNB methods have reduced precn, of 94.55%, 90.73%, 95.88%, 94.31%, 95.89%, 95.71%, 92.97%, and 98.80%, correspondingly. Similarly, based on $reca_l$ the MSDNet system has a greater $reca_l$, of 99.20% while the ANN, RF, GBM, RKPCA, SVM, NB, Ensemble Method and GNB methodologies have a lesser $reca_l$, of 96.02%, 92.66%, 93.79%, 96.78%, 91.90%, 91.57%, 93.72%, and 96.81%, respectively. Furthermore, based on $F1_{\text{score}}$, the MSDNet technique has a larger $F1_{\text{score}}$, of 99.26% while the ANN, RF, GBM, RKPCA, SVM, NB, Ensemble Method and GNB approaches have smaller $F1_{\text{score}}$, of 95.64%, 90.37%, 93.43%, 90.55%, 93.23%, 89.73%, 92.56%, and 97.40%, correspondingly.

Table 3. Comparative outcome of MSDNet technique with existing models

Techniques	$accu_y$	$prec_n$	$reca_l$	$F1$score
ANN algorithm	94.64	94.55	96.02	95.64
Random forest	89.80	90.73	92.66	90.37
GBM model	93.67	95.88	93.79	93.43
RKPCA technique	90.59	94.31	96.78	90.55
SVM classifier	98.03	95.89	91.90	93.23
Naive bayes	98.40	95.71	91.57	89.73
Ensemble method	97.93	92.97	93.72	92.56
GNB algorithm	98.65	98.80	96.81	97.40
MSDNet	99.27	99.33	99.20	99.26

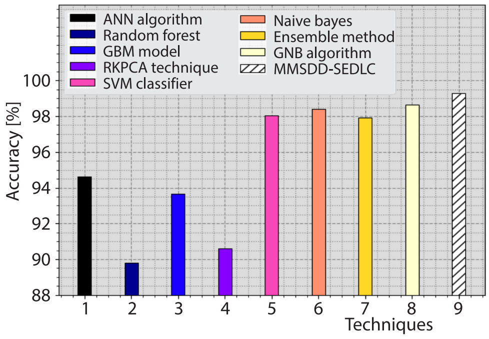

Figure 10. The $accu_y$ analysis of MSDNet technique with existing models

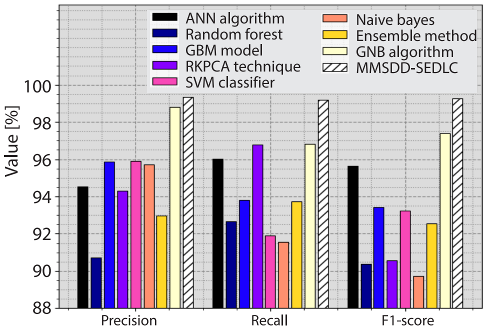

Figure 11. The $prec_n$, $reca_l$, and $F1_{\text{score}}$ analysis of MSDNet technique with existing models

In tab. 4 and fig. 12, the comparative results of the MSDNet technique are identified in terms of performance time (PT). The outcomes recommend that the MSDNet approach obtains improved performance. Based on PT, the MSDNet method delivers a lesser PT of 5.30 minutes whereas the ANN, RF, GBM, RKPCA, SVM, NB, ensemble method, and GNB systems attain greater PT values of 10.36 minutes, 9.15 minutes, 8.36 minutes, 13.65 minutes, 11.53 minutes, 9.86 minutes, 7.26 minutes, and 8.34 minutes, correspondingly.

Conclusion

In this article, we have introduced a novel MSDNet approach. The main objective of the MSDNet model is to recognize and classification of MS disease diagnosis. It comprises various kinds of stages involved as data normalization, feature selection, stacking ensemble of

Table 4. The PT outcome of MSDNet technique with recent approaches

Techniques	Performance time [minute]
ANN algorithm	10.36
Random forest	9.15
GBM model	8.36
RKPCA technique	13.65
SVM classifier	11.53
Naïve bayes	9.86
Ensemble method	7.26
GNB algorithm	8.34
MSDNet	5.30

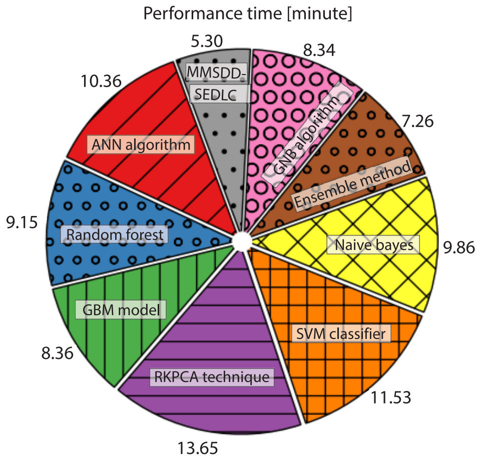

Figure 12. The PT outcome of MSDNet technique with other recent models

DL classifiers, and hyperparameter tuning processes. Firstly, the proposed MSDNet technique executes the min-max data normalization method to measure the data into a uniform range. Furthermore, an AHA-based feature selection process is taken to choose an optimal subset of features. For the MS diagnosis process, a stacked ensemble of DL classifiers is involved as LSTM, DNN, and CNN. Eventually, the https://genesys-academy-cs.slack.com/archives/D01CXJ64SRL/p1651296442330079EEWOA-assisted hyperparameter tuning method gets performed to enhance the efficiency of the stacking DL approaches. The experimentation outcome of the MSDNet approach undergoes and the performances are examined under altering aspects. The simulation study indicated the power of the MSDNet approach over existing models.

Data availability statement

The data that support the findings of this study are openly available in the Kaggle repository at https://www.kaggle.com/datasets/desalegngeb/conversion-predictors-of-cis-to-multiple-sclerosis [25].

Acknowledgement

The authors extend their appreciation to the Deanship of Scientific Research and Libraries in Princess Nourah bint Abdulrahman University for funding this research work through the Research Group project, Grant No. (RG-1445-0046)

References

[1] Coronado, I., *et al.*, Deep Learning Segmentation of Gadolinium-Enhancing Lesions in Multiple Sclerosis, *Mult. Scler. J.*, *27* (2018), 4, pp. 519-527

[2] Loizou, C. P., *et al.*, Normal Appearing Brain White Matter Changes in Relapsing Multiple Sclerosis: Texture Image and Classification Analysis in Serial MRI Scans, *Magn. Reson. Imaging*, *73* (2020), Nov., pp. 192–202

[3] Yoo, Y., *et al.*, Deep Learning of Brain Lesion Patterns and User-Defined Clinical and MRI Features for Predicting Conversion Multiple Sclerosis From Clinically Isolated Syndrome, *Comput. Methods Biomech. Biomed. Eng. Imaging Vis*. *7* (2019), 3, pp. 250-259

[4] Sacca, V., *et al.*, Evaluation of Machine Learning Algorithms Performance for the Prediction of Early Multiple Sclerosis from Resting-State Fmri Connectivity Data, Brain Imaging Behav, *13* (2019), 4, pp. 1103-1114

[5] Gaj, S., *et al*., Automatic Segmentation of Gadolinium-Enhancing Lesions in Multiple Sclerosis Using Deep Learning from Clinical MRI, *PLoS ONE*, *16* (2021), e0255939

[6] Narayana, P. A., *et al*., Deep Learning for Predicting Enhancing Lesions in Multiple Sclerosis from Non-Contrast MRI, *Radiology*, *294* (2020), 2, pp. 398-404

[7] Seok, J. M., *et al*., Differentiation between Multiple Sclerosis and Neuromyelitis Optica Spectrum Disorder Using A Deep Learning Model, *Sci. Rep*., *13* (2023), 11625

[8] Kavaklioglu, B.C., *et al*., Machine Learning Classification of Multiple Sclerosis in Children Using Optical Coherence Tomography, *Mult. Scler. J*., *28* (2022), 14, pp. 2253-2262

[9] Marzullo, A., *et al*., Classification of Multiple Sclerosis Clinical Profiles Via Graph Convolutional Neural Networks, Front, *Neuroscience*, *13* (2019), 594

[10] Ye, Z., *et al*., Deep Learning with Diffusion Basis Spectrum Imaging for Classification of Multiple Sclerosis Lesions, *Ann. Clin. Transl. Neurol*., *7* (2020), 5, pp. 695-706

[11] Yaghoubi, N., *et al*., Deep Learning and Classic Machine Learning Models in the Automatic Diagnosis of Multiple Sclerosis Using Retinal Vessels, *Multimedia Tools and Applications*, *83* (2024), 13, pp. 37483-37504

[12] Mohammed Aarif, K. O., *et al*., Exploring Challenges and Opportunities for the Early Detection of Multiple Sclerosis Using Deep Learning, in: *Artificial Intelligence and Autoimmune Diseases: Applications in the Diagnosis*, *Prognosis, and Therapeutics*, Springer, New York, USA, 2024, pp. 151-178

[13] NourEldeen, R. M., *et al*., Revolutionizing Neurological Diagnostics: Integrating 6G Technology with Deep Learning for Enhanced Detection of Multiple Sclerosis and Myelitis, *Proceedings*, IEEE, International Telecommunications Conference, Cairo, Egypt, 2024, pp. 602-608

[14] Langat, G., *et al*., Efficient Segmentation Model Using MRI Images and Deep Learning Techniques for Multiple Sclerosis Classification, *International Journal for Multiscale Computational Engineering*, *22* (2024), 5

[15] Iqbal, M. S., *et al*., Progress and Trends in Neurological Disorders Research Based on Deep Learning, *Computerized Medical Imaging and Graphics*, *116* (2024), 102400

[16] Ponce de Leon-Sanchez, E. R., *et al*., A Deep Learning Approach for Predicting Multiple Sclerosis, *Micromachines, 14* (2023), 4, 749

[17] Al Jannat, S., *et al*., Detection of Multiple Sclerosis Using Deep Learning, *Proceedings*, Asian Conference on Innovation in Technology ASIANCON, Pune, India, 2021, pp. 1-8

[18] Rode, K. N., Siddamallaiah, R. J., Image Segmentation with Priority Based Apposite Feature Extraction Model for Detection of Multiple Sclerosis in MR Images Using Deep Learning Technique, *Traitement du Signal*, *39* (2022), 2

[19] Kappal, S., Data Normalization Using Median Median Absolute Deviation MMAD-Based Z-Score for Robust Predictions vs. Min-Max Normalization, London Journal of Research in Science: *Natural and Formal*, *19* (2019), 4, pp. 39-44

[20] Sanampudi, A., Srinivasan, S., Local Search Enhanced Optimal Inception-ResNet-v2 for Classification of Long-Term Lung Diseases in Post-COVID-19 Patients, *Automatik*a, *65* (2024), 2, pp. 473-482

[21] Lazzarini, R., *et al*., A Stacking Ensemble of Deep Learning Models for IoT Intrusion Detection, *Knowledge-Based Systems*, *279* (2019.), 110941

[22] Zhou, S., *et al*., A Novel Malware Detection Model in the Software Supply Chain Based on LSTM and SVM, *Applied Sciences*, *14* (2024), 15, 6678

[23] Wang, S., *et al*., Modelling Non-Linear Shear Creep Behavior of A Structural Adhesive Using Deep Neural Networks (DNN), *Construction and Building Material*s, *414* (2024), 135083

[24] Amiri, A. F., *et al*., Fault Detection and Diagnosis of a Photovoltaic System Based on Deep Learning Using the Combination of a Convolutional Neural Network (CNN) and Bidirectional Gated Recurrent Unit (Bi-GRU), *Sustainability*, *16* (2024), 3, 1012

[25] ***, https://www.kaggle.com/datasets/desalegngeb/conversion-predictors-of-cis-to-multiple-sclerosis

[26] Aslam, N., *et al*., Multiple Sclerosis Diagnosis Using Machine Learning and Deep Learning: Challenges and Opportunities, *Sensors*, *22* (2022), 2, 7856

[27] Gharaibeh, M., *et al*., Optimal Integration of Machine Learning for Distinct Classification and Activity State Determination in Multiple Sclerosis and Neuromyelitis Optica, *Technologies*, *11* (2023), 5, 131

[28] Torkey, H., Belal, N. A., An Enhanced Multiple Sclerosis Disease Diagnosis Via an Ensemble Approach, *Diagnostics*, *12* (2022), 7, 1771

Paper submitted: June 20, 2024
Paper revised: September 5, 2024
Paper accepted: September 27, 2024