Parameter and Hyperparameter Optimisation of Deep Neural Network Model for Personalised Predictions of Asthma

Radiah Haque, Sin-Ban Ho, and Ian Chai
Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia
Email: {radumoni2, sinbanho}@gmail.com, ianchai@mmu.edu.my

Adina Abdullah
Faculty of Medicine, University of Malaya, Kuala Lumpur, Malaysia
Email: adina@ummc.edu.my

***Abstract*—Over the last couple of decades, numerous optimisation algorithms have been introduced to optimise machine learning models. However, until now, no evidence or framework can be found in the literature that adequately describes how to select the best algorithm for parameter and hyperparameter optimisation of the Deep Neural Network (DNN) model. In this paper, an enhanced Fragmented Grid Search (FGS) method has been introduced for tuning several hyperparameters and finding the optimal architecture of the DNN model using less computation power and time. Furthermore, several experimental models are trained on the asthma dataset using various optimisers to find the optimal parameters, which can help the DNN model converge towards the lowest loss value. The results show that the Adam optimiser provides the best accuracy rate (96%). Consequently, the optimised DNN model can be used for accurately providing personalised predictions of asthma exacerbations for effective asthma self-management.**

***Index Terms*—machine learning, deep neural networks, optimisation algorithm, personalization**

I. Introduction

Deep Neural Network (DNN) architecture allows optimisation and adjustment of its parameters (e.g. learning rate) and hyperparameters (e.g. hidden layers) before and during the training until the desired outcome is met and an optimised model is created [1]. It was observed that the DNN model is capable of providing robust performance when it is optimised [2]. This is because setting the optimal architecture and learning process of the DNN model helps to provide predictions with minimum errors and maximum accuracy. As such, optimising the DNN model can minimise the loss function (i.e. find the lowest loss value on a given dataset). However, this could be challenging and time consuming. To solve this issue, over the last couple of decades, various optimisation algorithms have been proposed which can automate the optimisation process and find the optimal neural architecture and training parameters. These algorithms can be divided into two primary groups: search-based optimisation and criteria-based optimisation algorithms, as seen in Fig. 1.

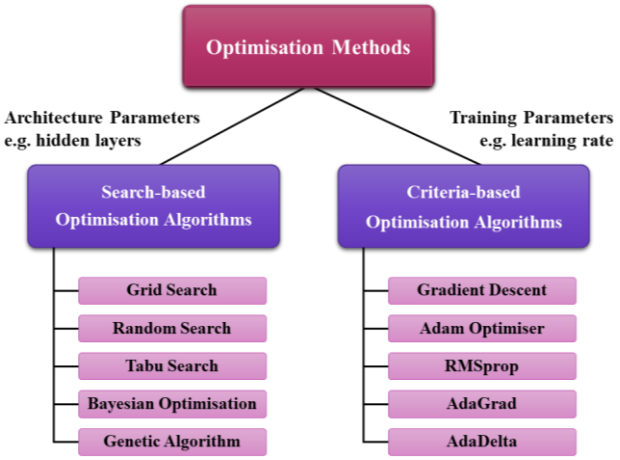

Figure 1. Optimisation methods of the DNN model.

The problem is, until now, there is no evidence or framework exists in the literature that adequately describes how to select the best optimisation algorithm for optimising the DNN model parameters and hyperparameters. Therefore, in this paper, several existing optimisation algorithms are reviewed and their limitations are identified. Subsequently, experimental procedures and comparative analyses are conducted to build an optimised regression-based DNN model for predicting personalised asthma exacerbations.

II. Search-Based Optimisation Algorithms

Search-based optimisation algorithms are responsible for finding the optimum values of the DNN architecture parameters (known as hyperparameters), which are set before training the model, such as hidden layers and hidden neurons. The procedure of finding optimal hyperparameters is called hyperparameter tuning [1].

Manuscript received January 5, 2022; revised March 15, 2022. accepted March 17, 2022.

doi: 10.12720/jait.13.5.512-517

Identifying the optimal hyperparameters of the DNN model is significant and considered an optimisation problem. Nonetheless, it should be noted that there is no standard formula or criterion for calculating the optimum values of the DNN hyperparameters.

Normally, for a shallow neural structure with limited hyperparameters, the tuning procedure can be done using the manual trial process. However, for a complex and deep neural structure with numerous potential hyperparameters, the tuning procedure is considered a combinatorial problem which is challenging and time-consuming [3]. In this regard, search-based optimisation algorithms are used which automate the trial procedure of tuning the hyperparameters and building the optimal architecture of the DNN model [4]. This happens by testing different potential hyperparameter values that exist in the search space through iterative trials, and thus selecting the best set of hyperparameter values for the model. As a result, the performance of the DNN model increases, which in turn gives the most accurate target outputs with less prediction errors [5].

Common optimisation algorithms include Grid Search (GS) [6], Tabu Search (TS) [3], Bayesian Optimisation (BO) [7], Random Search (RS) [8], and Genetic Algorithm (GA) [9]. High-level optimisation algorithms, such as BO and GA, can automate the trial procedure of hyperparameter tuning intelligently [5]. This is a significant limitation of the basic traditional optimisation algorithms, especially GS which requires a pre-defined search space set by the individual machine learning developer [6]. Having said that, GS provides more freedom than the other algorithms and offers the platform to test every possible combination of DNN hyperparameters and their potential values easily [10]. This is beneficial for testing the hyperparameter values that have been previously proven to perform well with other similar DNN models and prediction problems.

Meanwhile, it has been recognised that the main limitation of the majority of the existing optimisation algorithms is that they require prolonged execution time and need excessive computational power to test a large number of DNN hyperparameters and their potential values [11]. In this case, a key benefit of using GS is that it can be parallelised, which is useful to reduce time by parallelly applying the algorithm on multiple platforms. Nevertheless, there is a need to further explore the parallelisation feature of GS and identify its impact on hyperparameter tuning speed and time for optimising the DNN model. Hence, in this paper, a simple modified GS-based algorithm is proposed, which applies parallelisation along with an enhanced fragmentation method that can optimise the DNN model and tune several hyperparameters with less time and computational power.

III. Criteria-Based Optimisation Algorithms

Criteria-based optimisation algorithms (known as optimisers), are responsible for carrying out the learning process of the DNN model which takes place during its training [2]. The learning rate determines the step size for the DNN model that is required to minimise its loss function, thus achieving low error and high accuracy rates. This happens by ensuring that the model converges towards the lowest loss value possible efficiently [12]. Typically, the optimiser trains the DNN model in several iterations, known as epochs. At each epoch, the optimiser adjusts the parameters from the parameter space to decrease the loss value until the model converges towards the global minima. The optimiser often uses stopping criteria which makes the algorithm stop the iteration when a specific condition is met. Examples of common early stopping criteria include number of epochs, loss function value, and run-time. Fig. 2 shows how the parameters are optimised in the DNN model.

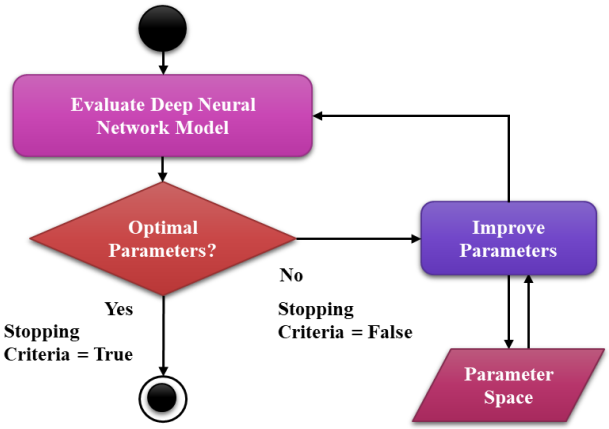

Figure 2. Optimisation process of the DNN model parameters.

Normally, when the loss function is convex, standard optimisers, such as Gradient Descent (GD), can find the optimal model parameters to converge towards the global minima [1]. Nevertheless, when the loss function is nonconvex, convergence to the global minima becomes difficult due to the existence of local regions with local minima. For the DNN model, the loss function is mostly nonconvex which could be due to the use of nonlinear activation functions. The Stochastic Gradient Descent (SGD) can handle nonconvex optimisation problems [2]. However, it has been identified that significant issues can arise when employing SGD with DNN, such as oscillation of the learning rate in the later training stages of the model. This can lead to increased variance and non-converging problems. Moreover, SGD tends to maintain a single learning rate in each epoch [13]. As a result, it might become difficult to identify the optimum learning rate value for training the DNN model.

In recent years, adaptive learning-based optimisers have been proposed to enhance both the learning rate and the convergence speed, such as Adaptive Gradient Descent (AdaGrad) [14], Root Mean Square Propagation (RMSProp) [15], and Adaptive Moment Estimation (Adam) [16]. Unlike SGD, the main property of AdaGrad is changing the learning rate for every parameter in each epoch [14]. One major issue of AdaGrad is that the learning rate keeps decreasing while training, which might result in vanishing gradients [17]. RMSProp solves this problem by using averaged squared gradients to normalise the gradients and avoid vanishing [15]. However, with RMSProp, the averaging process in each epoch can result

in repeated updating of the same learning rate, thus leading to a slower convergence and a longer optimisation process [13].

Fortunately, the Adam optimiser rectifies the gradient vanishing problem of AdaGrad using bounded gradients and the learning rate updating problem of SGD using adaptive learning rates. In addition, compared to RMSProp, Adam is known for its fast and smooth convergence while training the DNN model [16]. Therefore, Adam is a popular optimiser used in deep learning models because it can minimise the loss function efficiently. Furthermore, it was recognised that Adam can generate better prediction results than other adaptive learning-based optimisers on structured datasets, such as the asthma dataset [10]. Selected studies, however, have claimed that Adam might lead to worse generalisation performance than SGD. Nonetheless, it should be mentioned that the parameter spaces used in these studies are inadequate and incomparable because the optimum parameter values of the optimisers vary largely between datasets. In fact, to this date, there is no framework that adequately describes how to select the best optimiser for the DNN model based on the type of the dataset. Hence, in this paper, several experimental optimisation procedures are conducted and the comparative analyses of various optimisers are discussed. Consequently, the goal is to select the best optimiser for training the DNN model that can provide personalised predictions of asthma exacerbations with lowest errors and highest accuracy rates for effective asthma self-management.

IV. Personalised Predictions of Asthma

A. Asthma Dataset

The asthma dataset employed in this paper is collected from a mobile health application, known as Weather Asthma (WEA) [10]. The dataset includes weather variables (temperature, humidity, air pressure, wind speed, and UV index) and demography variables (age, gender, location, outdoor job, and outdoor activities) as the input features. The real-time values of the weather variables are collected from each location of individual WEA application users, which represent the weather conditions that trigger a particular user's asthma and lead to exacerbations. Meanwhile, the demography variables are collected from each user upon their registration in the WEA application, which are used for providing personalised predictions of asthma exacerbations based on weather triggers.

The asthma dataset also contains the Asthma Control Test (ACT) scores as the output variable, as seen in Table I. Users of the WEA application have been regularly conducting ACTs to report their asthma severity by answering five multiple choice questions. Each question scores from 1 to 5 and the total ACT scores can be from 5 to 25. According to the Global Initiative for Asthma [10], low ACT scores indicate high chances of asthma exacerbations and high ACT scores indicate low chances of asthma exacerbations. As such, each ACT score is stored in the dataset along with the demography information of the user who submitted the ACT, and the weather information of that day and time in that user's location. Consequently, the aim is to create an optimised DNN model that can recognise the patterns in the asthma dataset, predict the ACT scores based on their corresponding weather and demography variables, and offer personalised predictions of asthma exacerbations to WEA users based on daily weather forecasts.

TABLE I. Asthma Dataset Variables

No	Variable Name	Variable Type
1	ACT_score	Target Output
2	Temperature	Input (Weather Features)
3	Humidity	Input (Weather Features)
4	Pressure	Input (Weather Features)
5	Wind_speed	Input (Weather Features)
6	Location	Input (Demography Features)
7	Age	Input (Demography Features)
8	Gender	Input (Demography Features)
9	Outdoor_job	Input (Demography Features)
10	Outdoor_activities	Input (Demography Features)

B. Regression-Based DNN

DNN can be employed for modelling different ML techniques, such as regression and classification. Regression is a supervised learning technique which is responsible for quantifying and characterising the relationships between the input and the output variables. Regression is used for predicting numerical or continuous values. As such, in this paper, a regression-based DNN model is applied on the asthma dataset to predict the ACT scores.

Fig. 3 illustrates the predictive model architecture. The input layer contains ten input variables (x). The hidden layer contains n hidden layers and m number of hidden neurons at each hidden layer (h). The values of m and n are determined in the optimisation stage. The output layer contains one output variable (y). The rectifier (ReLU) is used as the activation function because of its ability to conduct simple nonlinear transformation of the input values in the hidden layers. The model assigns weights (w) to input values before passing them on to the hidden layer and output layer. Moreover, bias (b) is used which is a constant number that acts as an intercept value added for the purpose of adjustment. The following equations calculate the predicted values ($\hat{y}$) using the DNN model:

$$x = [x_1, x_2, \dots, x_{10}] \tag{1}$$

$$w = [w_1, w_2, \dots, w_{10}] \tag{2}$$

$$f(e) = \begin{cases} 0 & for \quad e < 0 \\ e & for \quad e \geq 0 \end{cases} \tag{3}$$

$$e = \sum_{i=1}^{10} xw + b \tag{4}$$

$$\hat{y} = f(e) \tag{5}$$

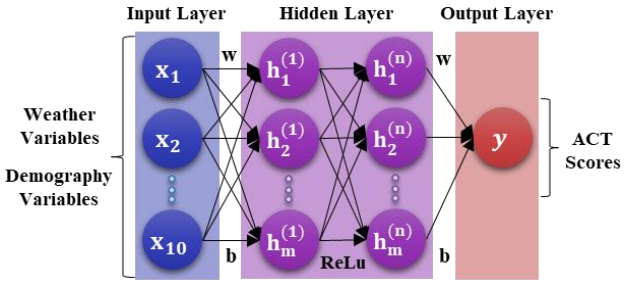

Figure 3. Regression-based DNN model architecture.

C. Evaluation Metrics

To evaluate the prediction performance of any regression model, error metrics are used as the evaluation metrics. Common error metrics include Mean Absolute Error (MAE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). The MAE is a basic loss function which sums up the absolute difference between the output values and the predicted values. This is achieved by measuring the median magnitude of the residuals from the prediction result equally. The MSE is a common loss function used for regression models which sums up the squared differences between the actual values and the predicted values. This is achieved by measuring the average of the squares of the errors. The main difference between the MAE and the MSE is that the latter can measure the outlier predicted values in the dataset. Meanwhile, the RMSE is the square root of MSE. The main difference between the MSE and the RMSE is that the latter is measured with the same unit as the target values. The following equations calculate the MAE, MSE and RMSE of the DNN model:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \quad (6)$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \quad (7)$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \quad (8)$$

where n is the number of records in the dataset, y is the actual output values, and $\hat{y}$ is the predicted values.

For further performance analysis, the overall accuracy rate and the Explained Variance Score (EVS) of the experimental models were calculated. The EVS indicates how well each model can capture the variance in the data that exists in the nonlinear dataset. The following equation calculates the EVS of the DNN model:

$$EVS = 1 - \frac{Var(y-\hat{y})}{Var(y)} \quad (9)$$

where v is the biased variance, y is the actual output values, and $\hat{y}$ is the predicted values.

V. Optimisation Results and Analysis

A. Hyperparameter Optimisation

In this paper, a Fragmented Grid Search (FGS) method is proposed for tuning the hyperparameters and finding the optimal DNN model architecture. FGS is a GS-based optimisation algorithm which utilises the parallelisation feature of GS and combines it with the fragmentation method [10]. In FGS, each hyperparameter is tuned independently along with its potential values, and the optimum value obtained from tuning one hyperparameter is used to tune the subsequent hyperparameter/s. The search space in each hyperparameter tuning experiment includes several potential values that have been commonly used with DNN and deep learning-based models. Algorithm 1 demonstrates the steps of FGS for tuning the DNN hyperparameters. Consequently, five DNN hyperparameters were tuned using the FGS algorithm, which include number of hidden layers, number of hidden nodes at each hidden layer, batch size, number of epochs, and type of the weight initialiser. Interestingly, the overall tuning process time of all five hyperparameters took around only 26 minutes in total. This is significantly faster than using GS, which takes 4 hours to tune only two hyperparameters [11]. Table II shows the optimum hyperparameter values used for building the regression-based DNN model architecture.

ALGORITHM 1: Fragmented Grid Search (FGS)

```
consider hyperparameter P
consider P candidate C
for each P, do
    create DNN model
    create list of P candidates in search space
    for each C in the list of P candidates, do
        train DNN model with C
        calculate model MSE
        if MSE < MinMSE, do
            set MinMSE = MSE
            set BestC = C
    return MinMSE
    return BestC
repeat for next P using previous BestC
```

TABLE II. Optimum Hyperparameters of DNN Model

No	Hyperparameter	Search Space Values	Optimum Value
1	Hidden Layers	1, 2, 3, 4, 5	2
2	Hidden Nodes	20, 30, 40, 50, 60	50
3	Bach Size	5, 10, 50, 100	10
4	Epochs	50, 100, 200, 1000	100
5	Weight Initialiser	Normal, Uniform	Normal

B. Parameter Optimisation

Several models were trained with the asthma dataset using various optimisers, including SGD, Adam, RMSProp, AdaDelta, and AdaGrad. The optimum number of epochs was used as the stopping criteria during each optimisation process. The dataset was split into a train set (1616 samples) for training and test set (404 samples) for validation. Fig. 4 and Fig. 5 compare the performance of the optimisers based on the training and test loss and evaluation metrics. It can be observed that the regression-based DNN model has achieved the best prediction results with the lowest error rates and the least training and validation loss using the Adam optimiser. It can also be recognised that the generalisation error is relatively low when the model is trained using Adam.

Furthermore, it can be seen that the Adam optimiser gives the best prediction results with the highest EVS and an accuracy rate of around 95% and 96% respectively. The experimental results concur with previous work [16], where it was recognised that Adam is an effective optimiser for training DNN models and produces better prediction results than other optimisers, such as SGD. Moreover, the findings reveal that training the DNN model on the structured dataset takes the least amount of time using the Adam optimiser. With this, the model converges towards the lowest loss value efficiently. Table III summarises the prediction results of each model training optimiser.

TABLE III. PERFORMANCE RESULTS OF THE DNN OPTIMIZERS

Optimiser	MAE	MSE	RMSE	EVS	Accuracy
SGD	0.23	0.10	0.31	0.91	92%
Adam	**0.13***	**0.05***	**0.22***	**0.95***	**96%***
RMSProp	0.16	0.06	0.25	0.94	95%
AdaDelta	0.18	0.08	0.27	0.93	94%
AdaGrad	0.25	0.11	0.34	0.89	91%

*Note: Bold fonts to show the best achieved results.

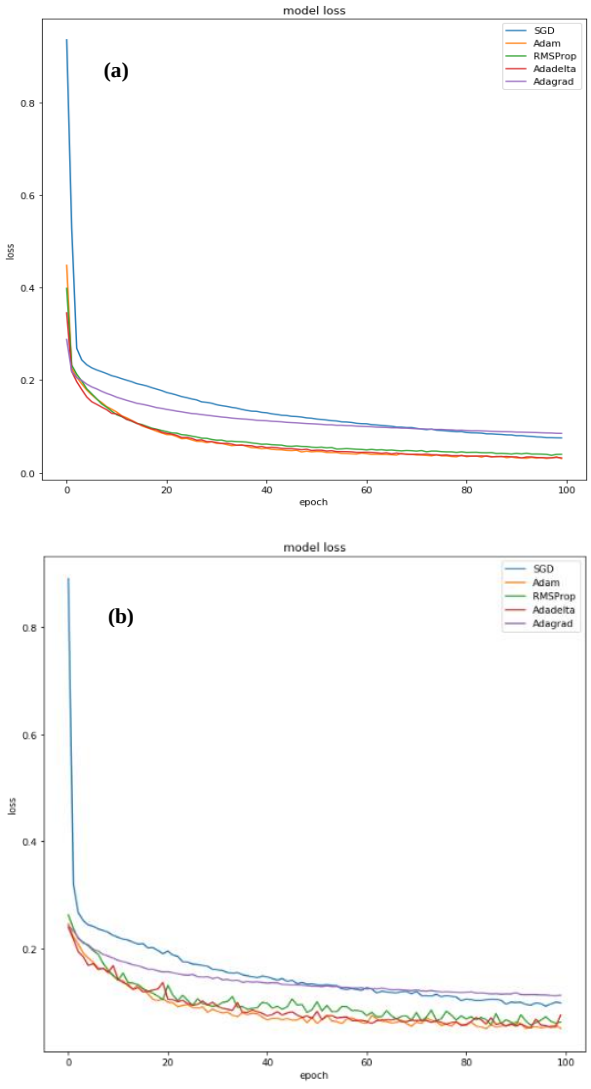

Figure 4. (a) comparison of training loss (Train MSE), (b) comparison of validation loss (Test MSE).

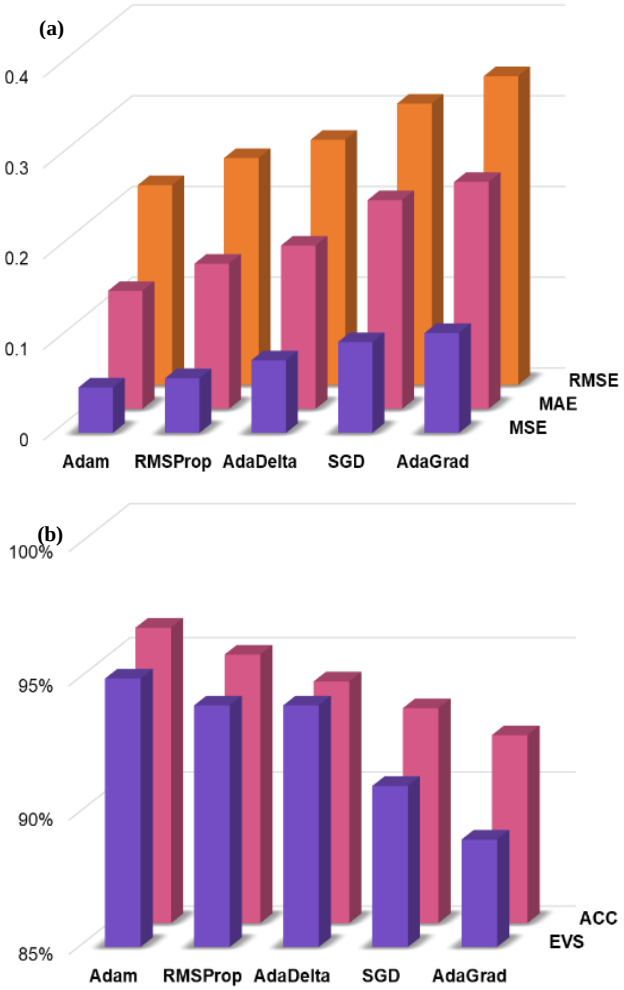

Figure 5. (a) comparison of error metrics, (b) comparison of variance and prediction accuracy.

VI. CONCLUSION

In this paper, several optimisation algorithms for optimising the DNN model parameters and hyperparameters have been reviewed. Two types of optimisation methods have been identified: search-based optimisation algorithms for hyperparameter tuning, and criteria-based optimisation algorithms (optimisers) for model training and updating the learning rates. Subsequently, a regression-based DNN model has been proposed and applied to the asthma dataset. To find the optimal model architecture, the FGS algorithm was used to tune several hyperparameters efficiently. Afterward, various experimental models were trained using common optimisers. Adam proved to outperform the other optimisers providing predictions with the lowest errors and highest accuracy rate. Moreover, the experimental results show that the Adam optimiser produces good generalisation performance when the model is trained on structured dataset (e.g. asthma dataset). Consequently, the optimised DNN model can be integrated to the WEA application to provide accurate personalised predictions of asthma exacerbations and offer tailored feedback to users for effective asthma self-management and weather-based healthcare.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

All the authors have contributed to this study equally.

ACKNOWLEDGMENT

The authors appreciate the financial supports provided for this study by the Fundamental Research Grant Scheme (FRGS/1/2019/SS06/MMU/02/4).

REFERENCES

[1] V. Alto. (2019). Neural networks: Parameters, hyperparameters and optimization strategies. *Towards Data Science*. [Online]. Available: https://towardsdatascience.com/neural-networks-parameters-hyperparameters-and-optimization-strategies-3f0842fac0a5

[2] K. Sankararaman, S. De, Z. Xu, W. Huang, and T. Goldstein, "The impact of neural network overparameterization on gradient confusion and stochastic gradient descent," in *Proc. 37th International Conference on Machine Learning*, Vienna, Austria, 2020, pp. 8469-8479.

[3] T. Gupta and K. Raza, "Optimizing deep feedforward neural network architecture: A tabu search based approach," *Neural Processing Letters*, vol. 51, no. 3, pp. 2855-2870, 2020.

[4] H. Mendoza, A. Klein, M. Feurer, J. Springenberg, M. Urban, and M. Burkart, "Towards automatically-tuned deep neural networks," *Automated Machine Learning*, pp. 135-149, 2019.

[5] A. Feitosa-Neto, J. Xavier-Júnior, A. Canuto, and A. Oliveira, "A study of model and hyper-parameter selection strategies for classifier ensembles: A robust analysis on different optimization algorithms and extended results," *Natural Computing*, vol. 20, pp. 805-819, 2021.

[6] G. Erten, S. Keser, and M. Yavuz, "Grid search optimised artificial neural network for open stope stability prediction," *International Journal of Mining, Reclamation and Environment*, vol. 35, no. 8, pp. 600-617, 2021.

[7] S. Greenhill, S. Rana, S. Gupta, P. Vellanki, and S. Venkatesh, "Bayesian optimization for adaptive experimental design: A review," *IEEE Access*, vol. 8, pp. 13937-13948, 2020.

[8] L. Li and A. Talwalkar, "Random search and reproducibility for neural architecture search," in *Proc. 35th Uncertainty in Artificial Intelligence Conference*, vol. 115, pp. 367-377, 2020.

[9] K. Hamdia, X. Zhuang, and T. Rabczuk, "An efficient optimization approach for designing machine learning models based on genetic algorithm," *Neural Computing and Applications*, vol. 33, no. 6, pp. 1923-1933, 2020.

[10] R. Haque, S. B. Ho, I. Chai, and A. Abdullah, "Optimised deep neural network model to predict asthma exacerbation based on personalised weather triggers," *F1000Research*, 2021.

[11] P. Liashchynskyi, "Grid search, random search, genetic algorithm: A big comparison for NAS," arXiv preprint arXiv:1912.06059, 2019.

[12] G. Perin and S. Picek, "On the influence of optimizers in deep learning-based side-channel analysis," in *Proc. International Conference on Selected Areas in Cryptography*, 2020, pp. 615-636.

[13] S. Sun, Z. Cao, H. Zhu, and J. Zhao, "A survey of optimization methods from a machine learning perspective," *IEEE Transactions on Cybernetics*, vol. 50, no. 8, pp. 3668-3681, 2019.

[14] A. Lydia and F. Francis, "Adagrad - An optimizer for stochastic gradient descent," *International Journal of Information and Computing Science*, vol. 6, no. 5, pp. 566-568, 2019.

[15] D. Xu, S. Zhang, H. Zhang, and D. Mandic, "Convergence of the RMSProp deep learning method with penalty for nonconvex optimization," *Neural Networks*, vol. 139, pp. 17-23, 2021.

[16] S. Sen and N. Ozkurt, "Convolutional neural network hyperparameter tuning with Adam optimizer for ECG classification," in *Proc. Innovations in Intelligent Systems and Applications Conference*, 2020, pp. 1-6.

[17] I. Kandel, M. Castelli, and A. Popovič, "Comparative study of first order optimizers for image classification using convolutional neural networks on histopathology images," *Journal of Imaging*, vol. 6, no. 9, article 92, pp. 1-17, 2020.

Radiah Haque received the B.S. degree in information and communication technologies from the Open University, United Kingdom, in 2013, and the M.S. degree in information technology with distinction from Coventry University, United Kingdom, in 2016. She is currently pursuing the Ph.D. degree in information technology at Multimedia University, Cyberjaya, Malaysia.

Sin-Ban Ho received the B.Sc. and M.Sc. degrees in computer science from the University of Science Malaysia, in 1998 and 1999, and the Ph.D. degree in information technology from Multimedia University, Cyberjaya, Malaysia, in 2008. He is currently a senior lecturer in the faculty of computing and informatics at Multimedia University, Cyberjaya, Malaysia. He is a senior member of the IEEE and IEEE Computer Society.

Ian Chai received the B.S. degree in computer science in 1988, and the M.S. degree in computer science in 1991, both from the University of Kansas, US. He worked in Connecticut and Germany, then received the Ph.D. degree in computer science from the University of Illinois, United States, in 2000. He is currently a principle lecturer in the faculty of computing and informatics at Multimedia University, Cyberjaya, Malaysia.

Adina Abdullah received the B.Med.Sci and B.M.B.S from University of Nottingham, UK, in 1997 and 1999, and the masters of medicine (family medicine) from the University of Malaya, Malaysia, in 2010. She is currently an associate professor in the department of primary care medicine, faculty of medicine, at the University of Malaya, Malaysia.