IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 12, No. 3, September 2023, pp. 1191~1203
ISSN: 2252-8938, DOI: 10.11591/ijai.v12.i3.pp1191-1203

Neural network models selection scheme for health mobile app development

Yaya Sudarya Triana[1], Mohd Azam Osman[2], Adji Pratomo[1], Muhammad Fermi Pasha[3], Deris Stiawan[4], Rahmat Budiarto[1]
[1]System Information Department, Faculty of Computer Science, Universitas Mercu Buana, Jakarta, Indonesia
[2]School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia
[3]Malaysia School of Information Technology, Monash University, Bandar Sunway, Malaysia
[4]Computer Engineering Department, Faculty of Computer Science, Universitas Sriwijaya, Palembang, Indonesia

Article Info

Article history:

Received Dec 29, 2022
Revised Jan 17, 2023
Accepted Jan 30, 2023

Keywords:

Graphical user interface design
Healthcare mobile app
Machine learning
Neural network

ABSTRACT

Mobile healthcare application (mHealth app) assists the frontline health worker in providing necessary health services to the patient. Unfortunately, existing mHealth apps continue to have accuracy issues and limited number of disease detection systems. Thus, an intelligent disease diagnostics system may help medical staff as well as people in poor communities in rural areas. This study proposes a scheme for simultaneously selecting the best neural network models for intelligent disease detection systems on mobile devices. To find the best models for a given dataset, the proposed scheme employs neural network models capable of evolving altered neural network architectures. Eight neural network models are developed simultaneously and then implemented on the android studio platform. Mobile health applications use pre-trained neural network models to provide users with disease prediction results. The performance of the mobile application is measured against the existing available datasets. The trained neural network engines perform admirably, detecting 7 out of 8 diseases with high accuracy ranging from 86% to 100% and a low detection accuracy of 63%. The detection times vary from 0.01 to 0.057 seconds. The developed mHealth app may be used by health workers and patients to improve resource-poor community health services and patients' healthcare quality.

Corresponding Author:

Rahmat Budiarto
Faculty of Computer Science, Universitas Mercu Buana
Jakarta, Indonesia
Email: rahmat.budiarto@mercubuana.ac.id

1. INTRODUCTION

Rural residents generally have restricted access to medical services; thus, unnecessary and repetitive procedures appear, causing a significant delay in the treatment process [1]. In addition, medical professionals are not available daily and visit the health center only once a month. For instance, the inability of front-line medical staff to determine which patients needed immediate treatment Front-line medical staffs have to assign each patient to a queuing system based on the principle of "first come, first served." Severely ill patients have no choice but to wait their turn. A healthier population can increase a country's economic production and lead to an increase in gross domestic product (GDP) [2].

Braun *et al.* [3] reported that health workers are crucial because they are the frontline providers. Because they are the first healthcare staff to interact with the patient, frontline health workers will be able to provide necessary health services with the help of the application [4]. For instance, the health worker can provide necessary health information to the patient and keep the patient calm. Correspondingly, they could help patients understand their health conditions and offer first aid to the patient if required. They could also classify the patients into different categories based on the severity of their health conditions. When illnesses are diagnosed early, it may help resource-poor communities identify potential illnesses sooner and may speed up procedures and treatments in the medical center while raising community health awareness.

A lot of research has been carried out on many aspects of healthcare system development, for example, in the security aspect [5] and healthcare quality of service [6]. The presence of artificial intelligence (AI) technology is very helpful for health service providers in healthcare, so the paramedics may provide necessary initial treatments while waiting for medical specialists' arrival. It will be more cost-effective if done at an early stage. Medical diagnostic research has been conducted by Kaur *et al.* [7], Jiang *et al.* [8], and Walter *et al.* [9].

mHealth is a portable application that allows multiuser connectivity, information sharing and AI techniques adoption [10]. mHealth provided a brand-new model of healthcare that allows for remote health monitoring, health data research, easy documentation, and other benefits. It enables patients and clinicians to easily access Internet health resources. mHealth can also be considered an educational app, a patient information and documentation tool, a patient monitoring app, a decision support tool, and a patient support tool [11], [12]. Many mHealth apps already exist in Malaysia and Indonesia, including myHealth, myMaHTAS, iDengue, Halodoc, Alodokter, KlikDokter, Medi-Call and SehatQ. Nevertheless, it has been reported that there are limited implementations of mobile health applications in resource-poor settings in the region. The main challenge is the lack of a scheme for developing multi-model systems to perform simultaneous disease diagnosis with acceptable accuracy. The existing scheme is only able to produce one model. Furthermore, a suitable AI architecture model must be designed in order to be successfully implemented into Android Studio and to develop a functional mobile health application.

Hence, this study proposes a scheme for selecting neural network models that best fit the Health Mobile App. This study extends the work in [13], develops suitable neural network models scheme to perform multi diseases diagnosis. For a successful implementation in Android Studio and to develop a well-functioning mobile health application, a fit model needs to be designed and an appropriate user interface design needs to be considered as mobile devices have some limitations in terms of CPU, memory, and power. To perform disease diagnosis and be able to simultaneously build several models, machine learning techniques are adopted in the proposed scheme. The diagnosis systems produced by the scheme are trained using publicly available medical datasets to create the predictive models. Neural networks-based mobile applications can predict possible medical conditions based on patient input. Model selection is the process of selecting the final neural network model from a collection of candidate neural network models. Different training dataset models with the same type of hyperparameters. Many studies have proposed model selection methods and approaches [13], [14]. However, the proposed approach focuses on only one or two datasets. The proposed scheme uses an evolutionary algorithm capable of evolving altered neural network architectures to discover the best neural network models for a given dataset. Eight neural network models are developed and then implemented on the Android Studio platform. Mobile health applications use pre-trained neural network models to provide users with disease prediction results. The performance of the mobile application is measured against the existing available datasets.

The key contribution of this research is a scheme for simultaneously selecting the best neural network models for disease diagnosis systems on mobile devices. The specific contributions as part of the proposed scheme are,

- A novel disease diagnosis application that is able to provide disease prediction based on user inputs. The mHealth application is mainly targeting the health worker and patient.
- A platform to improve the health status of each individual in resource-poor settings by offering each patient an opportunity to better understand the relationship between symptoms and diseases.
- The mHealth application itself promotes patient involvement in their own health and proactive management of wellness enhancement in resource-limited settings.

2. THE PROPOSED METHOD

The biggest challenge for AI-based medical diagnostic systems is predictive accuracy [7]–[9]. One approach to improving prediction accuracy is to carefully choose the correct neural network model. Disease prediction studies were performed using different types of algorithms in the AI approaches, as summarized in Table 1.

The overall architecture of the proposed scheme is shown in Figure 1. The proposed scheme is able to construct several neural network models simultaneously according to the inputted datasets. The scheme consists of five components and n datasets. Each component is described in detail in the following sections.

Table 1. Selected works on diseases diagnositcs systems

System	Author(s) and Year	Technique used	Purpose and Finding
Diabetic Diagnostics	Liu (2020) [15]	Neural Network	- An AI diagnostic model for the MATLAB software platform based on a neural network propagated by data acquisition for input feature vector integration, extraction, and selection.
	Xiang *et al.* (2021) [16]	Random Forest	- The proposed technique combines analysis of the physical and physiological characteristics of patients used in Chinese herbal medicine with tongue and pulse fundus photography.
Cancers Diagnostics	Raju *et al.* (2022) [17]	CNN, Computer Assisted Dianosis (CAD)	- A research survey on colon polyp detection and victimization with many totally different approaches to screening modalities. Discuss the significance and benefits of the setup modality, such as colonoscopy, as well as the associated CAD system. The classification of tumors using the CNN is discussed.
Breast Cancer Diagnostics	Freeman *et al.* (2021) [18]	Artificial Intelligence	- Study of the accuracy of AI for breast cancer detection in mammography screening practice.
	McKenney *et al.* (2020) [19]		- Introducing an AI system that can surpass experts in breast cancer prediction.
Heart Diagnostics	Choi *et al.* (2020) [20]	Artificial Intelligence	- A diagnostic system for cardiology (AICDSS) was established by combining expert systems and machine learning approaches for knowledge acquisition, then extending the knowledge base to include heart failure diagnosis.
	Sugiyarto *et al.* (2021) [21]	Convolution Neural Network	- Classify heart disease based on phonocardiogram (PCG) signals using the convolutional neural networks (CNN). - Experimental results during the training show the levels of accuracy, sensitivity, and diagnostic specificity are 100%, 100%, and 100%, respectively, and during the testing, the levels of accuracy, sensitivity, and specificity are 85%, 80%, and 100%, respectively.
Skin Diagnostics	Jain *et al.* (2021) [22]	Artificial Intelligence	- Evaluation of artificial intelligence (AI)-based tools for diagnosing dermatologic conditions.
Pneumonia	Masad *et al.* (2021) [23]	Hybrid deep learning	- Propose a hybrid deep learning system that consists of a convolutional neural network (CNN) model with three classifiers, i.e., k-nearest neighbor (KNN), support vector machine (SVM), and random forest (RF), to detect pneumonia from chest X-ray images. - The proposed approach achieved very efficient performance with a short classification time.

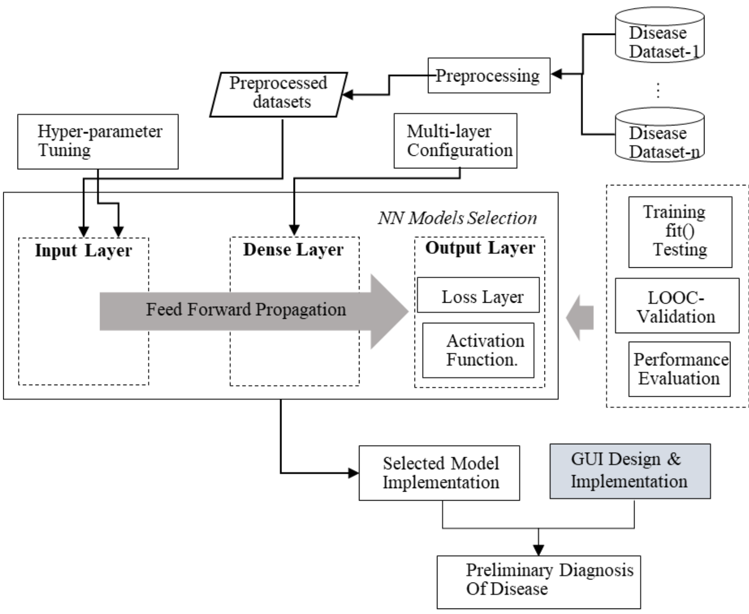


Figure 1. Overall architecture of the proposed scheme

2.1. Construction of the neural network model

Based on the structure of neurons in the human biological brain, a neural network model is built. Several layers of neurons contained in an artificial neural network (ANN) are connected either fully or partially. From neuron to neuron, the input signal will be transmitted, and each signal is represented by an input vector, X [24]. The postsynaptic potential function g was introduced to calculate the total number of signals affecting the neuron. In addition, the synaptic weight W is defined as the controller of synaptic transmission to neurons, i.e., the controller weighs the input signal at the synapse by decreasing or increasing the signal transmission. The number of multiplications can then be used by the researcher to determine whether the appropriate signal has reached the threshold required for neuronal activation.

The proposed scheme instantiates the multi layer configuration module to construct neural network models, and neural network hyperparameters are defined. By manipulating the weights of the model, the hyperparameters are used to orchestrate the learning process of the neural networks as well as produce an effective trained model. Two types of hyperparameters are commonly used, i.e., required and optional hyperparameters, as shown in Table 2.

Table 2. Required and optional hyperparameters

Required hyperparameters:	Optional hyperparameters:
Amount of input nodes	Learning rate schedule
Amount of hidden layers	Mini-batch size
Amount of output nodes	Momentum
Rate of Learning	Weight decay
Value of Weight	Dropout
Value of Bias	
Number of hidden nodes in each hidden layer	

The hidden layer of the neural network model uses feed-forward layers, i.e., a simple, fully connected dense layer. Then, for the output layer, the model uses the loss layer without parameters. Thus, the output layer consists of the loss function and activation function only. The neural network models follow forward propagation with two mathematical functions that are repeatedly used, i.e., summation and activation functions. The general summarization function is shown in (1).

$$\sum_{i=1}^{n} w_i X_i = 1 \tag{1}$$

where i is the index of summation, X_i is the set of values from input nodes, w_i their weights. Two activation functions, i.e., softmax function and hyperbolic tangent (tanh) function are used.

2.1.1. Neural network model selection algorithm

Underfitting and overfitting are the two common problems in designing neural network architecture. The use of a model with the most appropriate complexity is achieved by using the best generalization in order to produce a good data fit. This study adopts the model selection approach introduced by Laredo *et al.* [25]. The model selection approach uses an evolutionary algorithm capable of evolving altered neural network architectures to discover the best neural network model for a given dataset. The approach selects the evolutionary algorithm because it does not make any assumptions about the problem and just treats it as a black box that only provides quality measures of an obtained candidate model. Moreover, during the search for optimal models, the approach does not require implementing the gradient that is impossible to produce for a neural network. The pseudocode for the model selection is shown in Algorithm 1.

Algorithm 1: Model selection algorithm

```
Input: Training and cross-validation dataset for each disease.
 hyperparameters for the neural network (Refer to Table 2).
Output: The best neural network model for each disease.
Set count_e = 0 //experiments counter Create
While count_e < thresh_r do //thresh_r is a threshold
  Set count_g = 0 //the counter for generation.
  Create and initialize an initial population C(0), containing n individuals, n <= 20.
  While count_g < thresh_r do //nominal convergence not reached
    Check for nominal convergence in C(count).
    Evaluate the cost, cost(φ) of each candidate model φ.
    Identify best and worst models in C(count).
    Replace worst model in C(count) with the best from C(t − 1).
    Perform selection.
    Perform crossover of models in C(count) with ρ_c = 1.
    Set O(count) as the offspring population.
    For each model in O(count) perform mutation with probability ρ_m.
    Set C(count + 1) = O(count).
    count_g = count_g + 1.
    end
  Append the best solution from previous run to B.
  count_e = count_e + 1.
```

```
end
Return the best existing models in B as the best solutions
```

2.2. Datasets

Datasets for training, validation, and testing purposes are taken from the Machine Learning Repository of the University of California, Irvine [26]. We selected eight datasets of diseases commonly found in rural areas. Table 3 shows the selected datasets with their attributes (symptoms) and instances for each disease, as well as the reasons for the selection.

Table 3. Disease datasets' attributes and instances

Disease	Attrib. #	Inst. #	Rational for Selection
Acute Inflammation of Urinary Bladder	6	120	– It is a very common illness, with at least 60% of the female population. It is estimated that a patient have the disease [27]. – Easy diagnosis from medical condition and medical history.
Acute Nephritis	6	120	– The kidney is one of the most important parts of the human urinary system. – It can be diagnosed by physical examination, clinical examination, blood test, imaging test, and renal biopsy.
Cancer of Breast	32	569	– Breast cancer is the most common malignancy in women worldwide. – Early diagnosis and treatment may help patients survive breast cancer.
Disease of Skin	34	366	A fairly common illness that can be life-threatening.
Disease of Liver	6	345	– The liver is an important organ of the human body. – Early treatment reduces mortality.
Disease of Hepatitis	19	155	Same as Disease of Liver above
Disease of Heart	13	303	– Heart disease continues to be the leading murderer of death in developing countries – Therefore, the proposed health mobile application can serve as a reminder or warning to those experiencing the mentioned symptoms.
Disease of Thyroid	5	215	A fairly common illness. Hyperthyroidism is more common in females than in males.

Pre-processing

Records at the bottom of the datasets consist of blank lines. Some data labels are in the form of "strings." Therefore, record deletion and replacement of all non-integer entries with integers are required. Apache Spark and DataVec are utilized for this type of preprocessing.

The datasets are randomly divided into two parts: training datasets and testing datasets. Training datasets are used to customize the model. For the purposes of training the model, the model.fit () method is used to configure the model and pass the training data as arguments. The passing of the training data is repeated several times to find the right one. The testing datasets are used to evaluate the generalization error of the selected final model. Due to the limited data available for training and testing, the authors consider using leave-one-out cross validation (LOOCV) to validate the generated models to minimize bias. The whole datasets are used for model validation using the leave-one-out cross-validation method.

2.3. Leave-one-out cross validation (LOOCV)

LOOCV is a special case of the cross-validation approach where (*n-1*) records of data are considered the training dataset and one observation is used as the validation set. Since a model is repeatedly fitted to a dataset that contains *n-1* records of data, LOOCV produces a considerably less biased measure of test mean square error (MSE) compared to using a single test set. The LOOCV is implemented using Scikit-learn Python machine learning library via the LeaveOneOut class. The steps of the LOOCV approach are [28]:

- Split a dataset into a training dataset (n-1 observations) and a testing dataset (one left out observation)
- Generate the model based on the training dataset
- Calculate the prediction value for the one observation left out of the model
- Calculate the MSE (data value – prediction value)
- Repeat the process n times, use different observation as the leaving out from the training set.
- Repeat the process for each model.

The mean square error of the validation (MSE_v) is calculated using (2),

$$MSE_V = \frac{1}{n} * \sum_1^n MSE_i \qquad (2)$$

where:

MSE_i : The value of *MSE* during the i^{th} time of fitting the model.

n : The total number of records in the dataset of each model.

3. EXPERIMENTAL RESULTS AND DISCUSSION

The scheme is implemented using the Python programming language and a neural network library on a high-end computer with the following specifications: CPU: Intel Core i7-9700K, 8 cores (4.9 GHz), RAM: 16 GB DDR4, and 1 TB SSD storage, all running Microsoft Windows 10. In all experiments, the same dropout rates of 50% for all hidden units and 25% for visible units are used. The three different learning rates (= 0.1, 0.01, and 0.001) and the momentum for stochastic gradient descent (SGD) = 0.95, are also used.

3.1. The obtained neural network models

The selected optimal neural network models are saved and then used as pre-trained models in the Android Studio. Then, experiments using testing and validation data are carried out. Table 4 displays the three important hyperparameters for each optimal model.

3.2. Learning results

Figure 2 and Figure 3 show the accuracy and loss for the best selected models, respectively. Model 1, Model 2, and Model 7 achieve the highest accuracy of 100%. Model 6 provides the lowest accuracy of 63.39%. Model 1 converges fastest after 17 epochs. The lowest loss rate is achieved by Model 1, which is less than 4%, followed by Model 3 (less than 5.2%), and Model 2 and Model 7 (less than 6%). The rest of the models have losses greater than 10%.

3.3. Validation results

Table 5 shows the LOOCV results for the best selected models. Overall accuracy is good enough except for Model-6 (heart disease detection). Three models achieve hundred percent accuracy levels.

3.4. Performance measurement results

During the experiment, the confusion matrix values, i.e., true positive (TP), false negative (FN), false positive (FP), and true negative (TN) are recorded. These values are used for measuring other performance indicators: Precision, recall, and F1 score. Table 6 shows the confusion metrics results for the experiments on the five datasets.

Table 4. The trained models along with the best hyperparameters

Model	Detected Disease	# of Hidden Layer	η	Activation Function
Model-1	Acute Inflammation of Urinary Bladder	3	0.01	Tanh
Model-2	Acute Nephritis	3	0.001	Softmax
Model-3	Cancer of Breast	3	0.01	Tanh
Model-4	Disease of Skin	4	0.01	Tanh
Model-5	Disease of Hepatitis	4	0.001	Softmax
Model-6	Disease of Heart	3	0.001	Softmax
Model-7	Disease of Liver	3	0.001	Softmax
Model-8	Disease of Thyroid	3	0.001	Softmax

Table 5. LOOCV results

Model	Overall Accuracy	Overall MSE	Standard Deviation
Model-1	1.0000	0.0034	0.0099
Model-2	1.0000	0.0404	0.0097
Model-3	0.8800	0.0189	0.0306
Model-4	0.9700	0.0366	0.0586
Model-5	0.8600	0.0111	0.0195
Model-6	0.6300	0.0872	0.0813
Model-7	1.0000	0.0030	0.0088
Model-8	0.9700	0.0215	0.0335

Table 6. Confusion matrix for the model for the five diseases

	Confusion Matrix Results			
Model	TP	FN	FP	TN
Model-1	26	0	0	0
Model-2	19	0	0	29
Model-3	23	2	4	12
Model-4	47	4	5	6
Model-5	69	0	0	69
Model-6	53	2	2	1
Model-7	69	0	0	0
Model-8	64	2	11	9

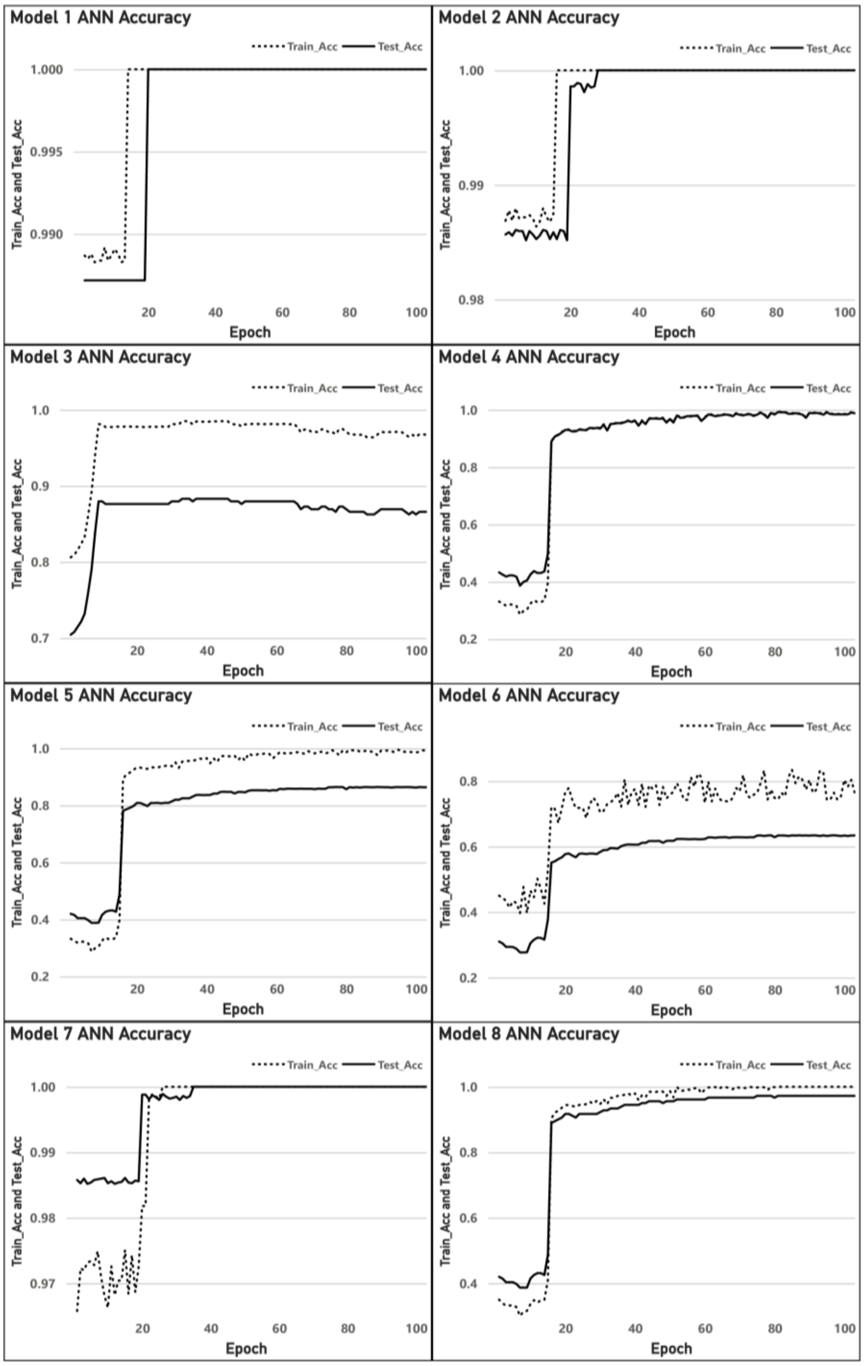


Figure 2. Training and testing accuracy

Three diseases, i.e., thyroid, heart, and skin disease, have two or more types. Table 7, Table 8, and Table 9 show the confusion matrices for thyroid, heart, and skin disorders, respectively. In Table 7, "0" means normal patients; "1" represents patients with hyperthyroidism, and "2" represents patients with hypothyroidism. 0 in Table 8 means psoriasis patients; "1" represents a patient with seborrheic dermatitis. "2" represents a patient with lichen planus. "3" represents pityriasis pink; "4" represents patients with chronic dermatitis. "5" represents a patient with lichen erythematosus. In Table 9, "0" represents patients with psoriasis. "1" represents a patient with seborrheic dermatitis. "2" represents a patient with lichen planus. "3" represents pityriasis pink; "4" represents patients with chronic dermatitis. "5" represents a nasal lichen patient. Table 10 shows the results of performance measurements from the experiments, i.e., accuracy, precision, recall and F1 score.

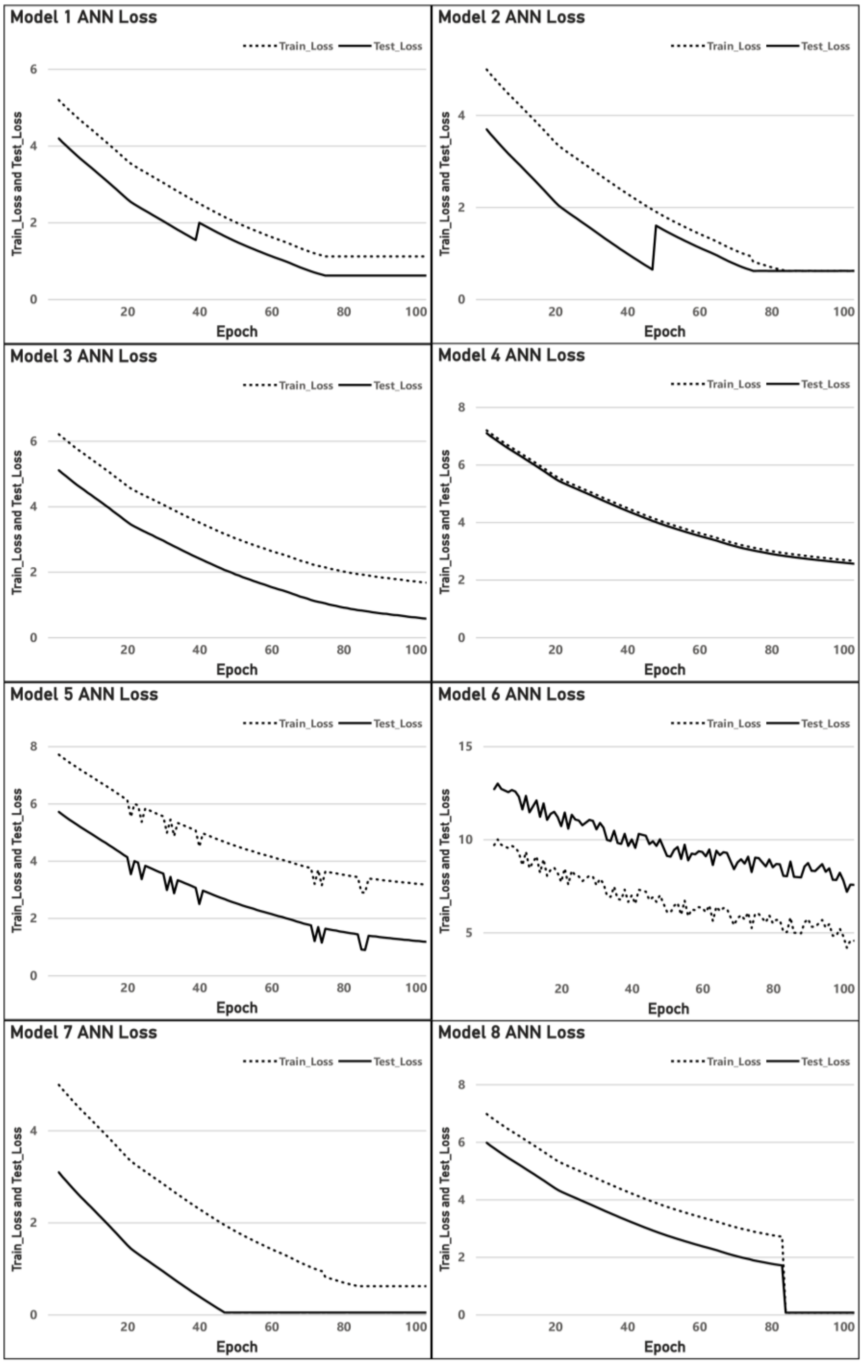


Figure 3. Training and testing loss

Table 7. Thyroid disease confusion matrix

	0	1	2
0	64	0	2
1	0	11	0
2	0	0	9

3.5. Comparison

The results of this work are compared to a work by Jerop and Segera [29] that combines principal component analysis (PCA) as feature selection, a hybrid kernel-based support vector machine (HKSVM) classification model, and hyperparameter optimization using a genetic algorithm (GA). The proposed HKSVM combines three kernels: linear, polynomial, and radial basis function (RBF), with the aim of improving accuracy performance. The comparison of the cross-validation accuracy is presented in Table 11.

Table 8. Heart disease confusion matrix

	0	1	2	3	4
0	53	2	2	1	1
1	9	3	1	7	0
2	2	2	2	7	1
3	3	0	1	8	0
4	0	0	0	1	0

Table 9. Skin disease confusion matrix

	0	1	2	3	4	5
0	41	0	0	0	0	
1	0	25	0	1	0	0
2	0	0	28	0	0	0
3	0	3	0	22	0	0
4	0	0	0	0	19	0
5	0	0	0	0	0	8

Table 10. Other performance measurement results

Disease	Performance Measurement Accuracy	Precision	Recall	F1 Score	Ave. Elapse time (seconds)
Acute Inflammation of Urinary Bladder (Model-1)	1	1	1	1	0.012
Acute Nephritis (Model-2)	1	1	1	1	0.011
Cancer of Breast (Model-3)	0.8865	0.8110	0.8617	0.81	0.049
Disease of Skin (Model-4)	0.9719	0.9688	0.9724	0.9740	0.057
Disease of Hepatitis (Model-5)	0.8642	0.5765	0.597	0.5983	0.011
Disease of Heart (Model-6)	0.6339	0.3867	0.3719	0.3415	0.027
Disease of Liver (Model-7)	1	1	1	1	0.024
Disease of Thyroid (Model-8)	0.9772	0.9391	0.9889	0.9618	0.010

Table 11. Cross validation accuracy comparison

Disease	Cross Validation Accuracy PCA-HKSVM	Proposed Scheme
Acute Inflammation of Urinary Bladder (Model-1)	0.98.97	1
Acute Nephritis (Model-2)	1	1
Cancer of Breast (Model-3)	0.9494	0.8865
Disease of Skin (Model-4)	N/A	0.9719
Disease of Hepatitis (Model-5)	N/A	0.8642
Disease of Heart (Model-6)	0.8097	0.6339
Disease of Liver (Model-7)	N/A	1
Disease of Thyroid (Model-8)	N/A	0.9772
Disease of Respiratory	0.9631	N/A
Acute Kidney	0.9594	N/A
Lympography	0.8644	N/A

3.6. Discussion

As shown in Table 4, only the neural network model for heart and skin diseases requires 4 hidden layers, and the rest require 3 hidden layers. The criterion for choosing the number of hidden layers is the number of classified diseases. ANNs with three hidden layers do not consume processing time for training as well as for classifying a disease with given symptoms.

Observing the experimental results in Table 5, the overall MSE for each model is relatively good, which means the models are unbiased. However, as expected, the variances are significantly large, as the LOOCV has a disadvantage, i.e., a high variance in prediction results. Nevertheless, for disease detection, classification, and prediction, an unbiased result is more important; thus, the generated models are acceptable. The confusion matrix in Table 6 shows relatively high true positive rates (TPR) and low false positive rates (FPR) for each model. In other words, the models are able to detect or predict the diseases.

Finally, the performance measurement results shown in Table 10 support the argument that the trained neural network engines perform well (with good accuracy) in detecting 7 out of 8 diseases and are appropriate for deployment on mobile health apps. The worst indicator is detecting heart disease. Additional work is required to improve the performance of the detection engine. The comparison of cross-validation accuracy results shows that the PCA-HKSVM scheme [27] performs slightly better than the proposed scheme for producing disease detection engines. Nevertheless, the PCA-HKSVM scheme is not suitable for adoption in mobile applications since it uses a combination of three methods, i.e., feature selection, classifier, and optimizer, which require more computing resources and computational processing time.

3.7. Mobile healthcare app implementation

The user interface design of any mobile application is important, as the interface should consider how people control and use the system in addition to how the system itself works. Therefore, proper user interface design is required to meet basic user requirements. Buttons in the mobile app interface are an example of *affordance*. Buttons represent potential uses and actions that the user can take when completing a specific task [30]. For example, users can easily understand that there are two buttons on the screen: The previous and next buttons. It is also important to provide *signifier*, i.e., information about the actions taken when users click the button with act as symbol, using words highlighted at the top of each button. Therefore, *signifiers* directly improve the provision of objects. If the user can understand the function of the button from the design of the application, it will be easier to use. Figure 4 shows an implementation example of *affordances* and *signifiers* in a mobile application.

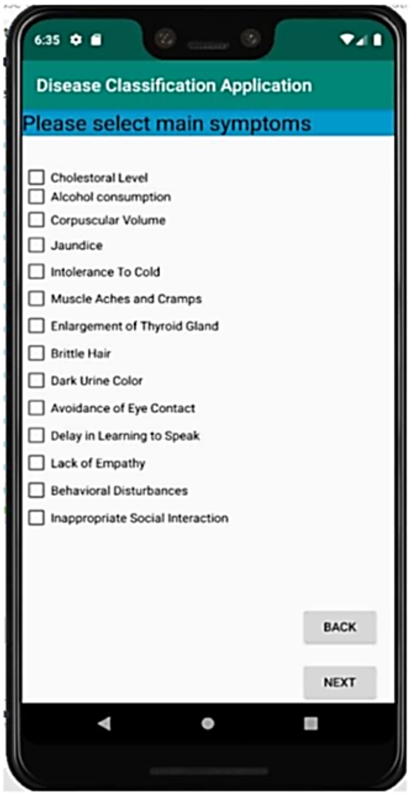


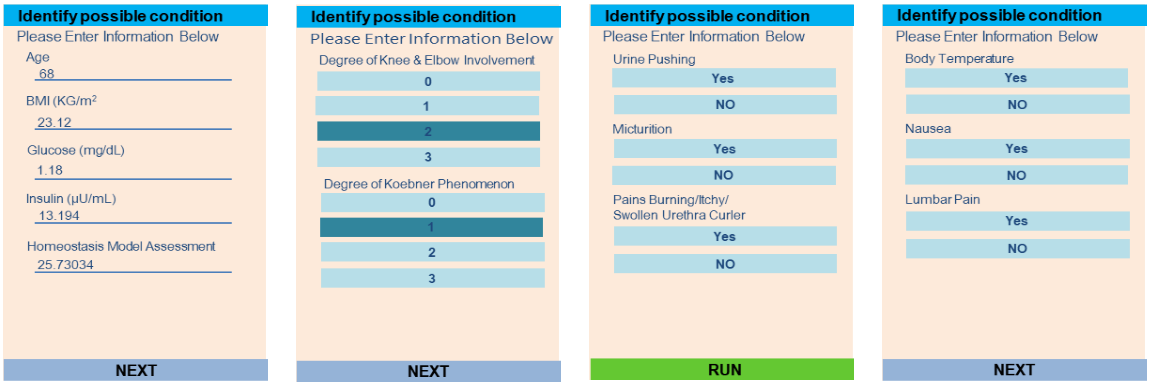
Figure 4. An example of affordance and signifier in the mobile application

As Deniz and Durdu [30] suggest, visual confusion must be reduced to avoid unnecessary information. One of the patient evaluation forms is shown in Figure 5(a), however, it is not cluttered with unnecessary information and it is designed to include only the required input in the patient review form. Nevertheless, this restriction does not prevent users from completing the required evaluation.

Another best practice is to use vertical characters instead of horizontal characters. Vertical lettering helps to take up too much space and makes it much easier to fit into the user interface. There is a limit to the mobile screen size. Therefore, vertical labeling is much better than horizontal labeling. According to Shiffman *et al.* [31], the author states that horizontal text increases unnecessary eye movements because users may have to move their eyes from left to right and from top to bottom. Therefore, the user interface is designed by practicing vertical markings that make it easier for users to view the rating form. We can reduce eye movement by requiring the user to move their eyes from top to bottom when labeling vertically. It indirectly reduces eyes strain and leads to faster scans compared to horizontal labeling. Figure 5(b) shows an example of vertical labeling.

In addition, the entire user interface of the mobile health application has been properly tuned. It is important to have a consistent and correct orientation on all screens, as incorrect orientation tends to frustrate users. In addition, properly placing elements in any user interface will improve the visual connection between related elements. In Figure 5(a), it is shown that all labels are left-aligned because that is the way people used to read (left-to-right, top-to-bottom). Proper placement ensures a good user experience by making it easier for users to complete the evaluation.

Finally, designing the user interface needs to be more careful with color choices. Different colors will have different meanings for different users [32]. Green is used because it can get the user's attention as shown in Figure 5(c). Overall, green has a symbolic meaning to humans. Blue is used as a contrast to the background color and usually has a neutral symbolic meaning as shown in Figure 5(d). In conclusion, color is important in user interface design and should be used properly to create a good interface.



(a) (b) (c) (d)

Figure 5. Other considerations in mobile application graphical user interface (GUI) design: (a) consistent and correct orientation, (b) vertical labeling, (c) and (d) careful color choice

4. CONCLUSION

This study has proposed a model selection scheme for simultaneously developing neural network models for an intelligent disease diagnostics system. The disease-diagnosis application is then implemented in Android Studio. Our aim is to provide an application that overcomes all limitations in a resource-poor environment. We are confident that new applications for diagnosing illness will enable patients to be more involved in their own health. It can also ensure that individuals are aware of information about their health, regardless of geographical barriers. New illness diagnostic applications can save users' time and money by providing them with a faster way to get illness diagnoses. Finally, this preliminary research can emphasize early detection of illness and improve preventive management of well-being in resource-poor environments. The limitation of this research is the unavailability of a public disease dataset locally that may represent more specific diseases' symptoms. Since this research work uses publicly available datasets, the trained diagnosis system may not give as good accuracy as the current experimental results if applied to local datasets. On the other hand, even though this study only considers eight common diseases, the proposed scheme should be able to produce more models simultaneously when the datasets are available. For future work, the researchers plan to utilize other learning methods, including transfer learning, ensemble learning, and enforcement learning, that can be implemented to improve the selected neural network's performance. No field trials were performed in this study. Therefore, this research does not perform application evaluation. Without an application evaluation, this research study cannot assess the impact of a real application. Thus, assessment and verification of the correctness of the application using actual clinical diagnosis are also considered future work. For this purpose, the app should be translated into Bahasa Indonesia and Bahasa Melayu first, as the respondents are people who live in poor communities in rural areas of Indonesia and Malaysia.

ACKNOWLEDGEMENTS

This research was fully funded by the Research & Technology Directorate of Universitas Mercu Buana under the grant number: 02-5/1337/B-SPK/V/2022.

REFERENCES

[1] S. Suprapto, T. C. Mulat, and N. S. N. Lalla, "Nurse competence in implementing public health care," *International Journal of Public Health Science (IJPHS)*, vol. 10, no. 2, p. 428, Jun. 2021, doi: 10.11591/ijphs.v10i2.20711.

[2] T. Fernholm and A. Marshall, "Principles of economics," *Ekonomisk Tidskrift*, vol. 65, no. 3, p. 129, 1963, doi: 10.2307/3438528.

[3] R. Braun, C. Catalani, J. Wimbush, and D. Israelski, "Community health workers and mobile technology: A systematic review of the literature," *PLoS ONE*, vol. 8, no. 6, 2013, doi: 10.1371/journal.pone.0065772.

[4] A. White, D. S. K. Thomas, N. Ezeanochie, and S. Bull, "Health worker mHealth utilization: A systematic review," *CIN - Computers Informatics Nursing*, vol. 34, no. 5, pp. 206–214, 2016, doi: 10.1097/CIN.0000000000000231.

[5] S. Purwanti, B. Nugraha, and M. Alaydrus, "Enhancing security on E-health private data using SHA-512," *2017 International Conference on Broadband Communication, Wireless Sensors and Powering, BCWSP 2017*, vol. 2018-January, pp. 1–4, 2018, doi: 10.1109/BCWSP.2017.8272557.

[6] H. H. Purba, F. Debora, C. Jaqin, and H. Adiyatna, "Service quality analysis: An empirical study of customer satisfaction in a healthcare," *Jurnal Teknologi dan Manajemen*, vol. 19, no. 1, pp. 33–38, 2021, doi: 10.52330/jtm.v19i1.22.

[7] S. Kaur *et al.*, "Medical diagnostic systems using artificial intelligence (AI) algorithms: Principles and perspectives," *IEEE Access*, vol. 8, pp. 228049–228069, 2020, doi: 10.1109/ACCESS.2020.3042273.

[8] L. Jiang *et al.*, "Opportunities and challenges of artificial intelligence in the medical field: current application, emerging problems, and problem-solving strategies," *Journal of International Medical Research*, vol. 49, no. 3, 2021, doi: 10.1177/03000605211000157.

[9] W. Walter *et al.*, "How artificial intelligence might disrupt diagnostics in hematology in the near future," *Oncogene*, vol. 40, no. 25, pp. 4271–4280, 2021, doi: 10.1038/s41388-021-01861-y.

[10] C. P. Waegemann, "mHealth: History, analysis, and implementation," *M-Health Innovations for Patient-Centered Care*, pp. 1–19, 2016, doi: 10.4018/978-1-4666-9861-1.ch001.

[11] F. O. Isinkaye, S. O. Awosupin, and J. Soyemi, "A Mobile Based Expert System for Disease Diagnosis and Medical Advice Provisioning," *International Journal of Computer Science and Information Security (IJCSIS)*, vol. 15, no. 1, pp. 568–572, 2017, [Online]. Available: https://sites.google.com/site/ijcsis/.

[12] P. Ngor *et al.*, "Smartphones for community health in rural Cambodia: A feasibility study [version 1; referees: 1 approved, 2 approved with reservations]," *Wellcome Open Research*, vol. 3, 2018, doi: 10.12688/wellcomeopenres.13751.1.

[13] S. Tang and J. Wiens, "Model selection for offline reinforcement learning: Practical considerations for healthcare settings," 2021, [Online]. Available: http://arxiv.org/abs/2107.11003.

[14] M. F. Pasha, H. S. Lee, A. Widhiasi, R. Purba, A. Mansour, and R. Budiarto, "Neural nework-based mobile app framework to aid resource-poor setting community health," *2020 International Conference on Computing and Information Technology, ICCIT 2020*, 2020, doi: 10.1109/ICCIT-144147971.2020.9213727.

[15] Y. Liu, "Artificial intelligence-based neural network for the diagnosis of diabetes: Model development," *JMIR Medical Informatics*, vol. 8, no. 5, 2020, doi: 10.2196/18682.

[16] Y. Xiang *et al.*, "Artificial intelligence-based diagnosis of diabetes mellitus: combining fundus photography with traditional chinese medicine diagnostic methodology," *BioMed Research International*, vol. 2021, 2021, doi: 10.1155/2021/5556057.

[17] A. S. Narasimha Raju, K. Jayavel, and T. Rajalakshmi, "Intelligent recognition of colorectal cancer combining application of computer-assisted diagnosis with deep learning approaches," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 1, p. 738, Feb. 2022, doi: 10.11591/ijece.v12i1.pp738-747.

[18] K. Freeman *et al.*, "Use of artificial intelligence for image analysis in breast cancer screening programmes: Systematic review of test accuracy," *The BMJ*, vol. 374, 2021, doi: 10.1136/bmj.n1872.

[19] S. M. McKinney *et al.*, "International evaluation of an AI system for breast cancer screening," *Nature*, vol. 577, no. 7788, pp. 89–94, 2020, doi: 10.1038/s41586-019-1799-6.

[20] D. J. Choi, J. J. Park, T. Ali, and S. Lee, "Artificial intelligence for the diagnosis of heart failure," *npj Digital Medicine*, vol. 3, no. 1, 2020, doi: 10.1038/s41746-020-0261-3.

[21] A. W. Sugiyarto, A. M. Abadi, and S. Sumarna, "Classification of heart disease based on PCG signal using CNN," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 5, p. 1697, Oct. 2021, doi: 10.12928/telkomnika.v19i5.20486.

[22] A. Jain *et al.*, "Development and assessment of an artificial intelligence-based tool for skin condition diagnosis by primary care physicians and nurse practitioners in teledermatology practices," *JAMA Network Open*, vol. 4, no. 4, 2021, doi: 10.1001/jamanetworkopen.2021.7249.

[23] I. S. Masad, A. Alqudah, A. M. Alqudah, and S. Almashaqbeh, "A hybrid deep learning approach towards building an intelligent system for pneumonia detection in chest X-ray images," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 6, p. 5530, Dec. 2021, doi: 10.11591/ijece.v11i6.pp5530-5540.

[24] M. Flasinki, "Introduction to artificial intellegence," *Springer International Publishing*, 2016.

[25] D. Laredo, Y. Qin, O. Schütze, and J.-Q. Sun, "Automatic model selection for neural networks," 2019, [Online]. Available: http://arxiv.org/abs/1905.06010.

[26] A. Frank and A. Asuncion, "UCI machine learning repository," 2010, [Online]. Available: http://archive.ics.uci.edu/ml.

[27] M. S. Azeez and Z. M. Maki, "Diagnosis urine disease based on KNN algorithm and ANN," *International Journal of Engineering and Technology(UAE)*, vol. 7, no. 4, pp. 3367–3371, 2018, doi: 10.14419/ijet.v7i4.13686.

[28] F. Marchetti, "The extension of Rippa's algorithm beyond LOOCV," *Applied Mathematics Letters*, vol. 120, 2021, doi: 10.1016/j.aml.2021.107262.

[29] B. Jerop and D. R. Segera, "An efficient PCA-GA-HKSVM-based disease diagnostic assistant," *BioMed Research International*, vol. 2021, 2021, doi: 10.1155/2021/4784057.

[30] G. Deniz and P. O. Durdu, "A comparison of mobile form controls for different tasks," *Computer Standards and Interfaces*, vol. 61, pp. 97–106, 2019, doi: 10.1016/j.csi.2018.06.001.

[31] R. N. Shiffman, Y. Liaw, C. A. Brandt, and G. J. Corb, "Computer-based guideline implementation systems: A systematic review of functionality and effectiveness," *Journal of the American Medical Informatics Association*, vol. 6, no. 2, pp. 104–114, 1999, doi: 10.1136/jamia.1999.0060104.

[32] M. F. Kamaruzaman, N. M. Rani, H. M. Nor, and M. H. H. Azahari, "Developing user interface design application for children with autism," *Procedia - Social and Behavioral Sciences*, vol. 217, pp. 887–894, 2016, doi: 10.1016/j.sbspro.2016.02.022.

BIOGRAPHIES OF AUTHORS

Yaya Sudarya Triana received B.Sc. degree in Statistics from University of Padjadjaran, Indonesia in 1988, M.Kom. in Master of Information Technology from University of Indonesia in 2002 and Ph.D. in Statistics from Universiti Malaysia Trengganu, Malaysia in 2013, Software Engineer 1991-2003 at an IT Company at Tokyo, Bandung & Jakarta, respectively. Currently, he is an Assistant Professor at Dept. of Information System, Universitas Mercu Buana Indonesia. His research interests include Fuzzy, Data Science, Artificial Intelligent, Business Intelligent, Data Analytics, Statistics, Information System, and Machine Learning. He can be contacted at email: yaya.sudarya@mercubuana.ac.id.

Mohd Azam Osman is an Assoc. Prof. at the School of Computer Sciences, Universiti Sains Malaysia. He obtained Bachelor degree and a Master degree in Computer Science from Universiti Sains Malaysia, Pulau Pinang, Malaysia in 1996 and 2000 respectively. He joined the School in 1996 as an information system officer and was appointed as a lecturer in 2000. He has taught programming, operating system, computer network and database courses at undergraduate and postgraduate levels. He has held administrative positions as First Year Coordinator and Computing Science Programme Manager. He has led more than 8 research grants and published more than 50 papers in journals and conference proceedings. His current research focuses on mobile applications, image processing and deep learning to detect and track multi-fish in underwater and human emotional identification. He can be contacted at email: azam@usm.my.

Adji Pratomo is currently a final year undergraduate student at Department of Informatics, Faculty of Computer Science, Universitas Mercu Buana, Jakarta, Indonesia. His researches interests include Artificial Intelligent, Data Science, Food Sustainability, Agriculture, and Animal Husbandry. He can be contacted at email: 41518210036@student.mercubuana.ac.id.

Muhammad Fermi Pasha was born in Indonesia. He received the Ph.D. degree in computer science from Universiti Sains Malaysia, in 2010. After his Ph.D. degree, he worked as a Research Fellow at Universiti Sains Malaysia. He is currently working as Senior Lecturer at the School of Information Technology, Monash University Malaysia. His research interests are focused on computational neuroimaging, intelligent network security trafic analysis, and healthcare and radiology IT with emphasis on big data. He is also supervising the Ph.D. students in the latter mentioned research areas. He can be contacted at email: muhammad.fermipasha@monash.edu.

Deris Stiawan received a Ph.D. degree in Computer Engineering from Universiti Teknologi Malaysia, Malaysia. He is currently an Associate Professor at the Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya. His research interests include computer networks, intrusion detection/prevention systems, heterogeneous networks, and intelligent systems. He can be contacted at email: deris@unsri.ac.id.

Rahmat Budiarto received B.Sc. degree in Mathematics from Bandung Institute of Technology, Indonesia in 1986, M.Eng. and Dr.Eng. in Computer Science from Nagoya Institute of Technology, Japan in 1995 and 1998, respectively. Currently, he is a full professor at Dept. of Informatics, Universitas Mercu Buana Indonesia. His research interests include intelligent systems, brain modeling, IPv6, network security, Wireless sensor networks, and MANETs. He can be contacted at email: rahmat.budiarto@mercubuana.ac.id.