

Single nucleotide polymorphism based on hypertension potential risk prediction using LSTM with Adam optimizer

Lailil Muflikhah¹, Imam Cholissodin¹, Nashi Widodo², Feri Eko Hermanto³,
Teresa Liliana Wargasetia⁴, Hana Ratnawati⁴, Riyanarto Sarno⁵

¹Department of Informatics Engineering, Faculty of Computer Science, Brawijaya University, Malang, Indonesia

²Faculty of Mathematics and Natural Sciences, Brawijaya University, Malang, Indonesia

³Faculty of Animal Sciences, Brawijaya University, Malang, Indonesia

⁴Faculty of Medicine, Maranatha Christian University, Bandung, Indonesia

⁵Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia

Article Info

Article history:

Received Oct 4, 2023

Revised Nov 27, 2023

Accepted Dec 17, 2023

Keywords:

Adam optimizer

Hypertension

LSTM

Prediction

SNP

ABSTRACT

Recent healthcare research has focused a great deal of interest on using genetic data analysis to predict the risk of hypertension. This paper presents a unique method for accurately predicting the vulnerability to hypertension by utilizing single nucleotide polymorphism (SNP) data. We present a novel neural network design utilizing the adaptive moment (Adam) optimizer to describe the intricate temporal correlations in SNPs. The study used a dataset with carefully preprocessed SNP data from a broad cohort for model input. The long short-term memory (LSTM) network was methodically built and trained with hyper-parameter and fine-tuning using the Adam optimizer to converge on ideal weights. Our findings indicate encouraging predictive performance, highlighting the suggested methodology's usefulness in determining hypertension risk factors. The result showed that the proposed method achieved stability in the performance of 89% accuracy, 96% precision, 88% recall, and 92% F1-score. Due to its higher accuracy and greater predictive power, our SNP-based LSTM methodology is superior to the conventional machine learning method. By providing a novel framework that uses genetic data to predict the risk of hypertension, this research makes substantial contribution to the field of predictive healthcare. This framework helps with early intervention and customized preventative efforts.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Lailil Muflikhah

Department of Informatics Engineering, Faculty of Computer Science, Brawijaya University

Malang, Indonesia

Email: lailil@ub.ac.id

1. INTRODUCTION

Dysregulated blood pressure, also known as hypertension, is widely distributed among society around the globe. According to medical research, detecting hypertension early, changing one's lifestyle, and strictly controlling it can lessen its effects. Most people with hypertension do not display any signs, making it known as a "silent killer". Some may experience headaches, difficulty breathing, nosebleeds, dizziness, or heart palpitations. However, these symptoms are not unique to the medical condition and often don't appear until it has become severe [1]. There is still much to be discovered about detecting and monitoring hypertension, particularly in less developed nations where access to preventative healthcare services is more limited. The multi-factorial and polygenic mechanism underlying this disease caused some limitations in the prevention assessment [2], [3]. Several genes that contribute to the development of hypertension, and variations of individual genetics in the form of single nucleotide polymorphism (SNP) have a significant role

in the susceptibility to develop this disorder [4]. Up to now, certain allele of specific SNP variations from various genes has more association with the development of hypertension [5]–[21]. Although some preventive methods have been developed to diagnose the potential risk of hypertension earlier [22]–[24], the results may unsatisfactorily provide a persistent result due to the polygenic characteristic of this disease. Thus, a combination of several polymorphisms from several genes becomes a promising approach to developing an early diagnosis of hypertension risk factor.

Research related to hypertension is conducted from early detection to the provision of treatment for hypertensive patients using machine learning methods [25], [26]. Many researchers and practitioners have proposed different learning algorithms to detect the risk of hypertension using a computational approach as a machine learning method using any data type. The most data used in their research was clinical data and blood pressure [27]. However, the limitations of machine learning methods require knowledge of which predictors are involved. When dealing with a large number of independent variables, feature selection becomes necessary to achieve high-performance evaluation. Some authors developed a computational analysis to deal with the individual genetics data and achieved a favourable result. Due to the large number of image features extracted, more than 10,000 features, this method is required to employ the feature selection algorithm to determine the potential feature in classifier model construction for disease risk prediction [21]–[23]. However, no persistent method was developed to deal with the SNP data for predicting the risk factor of hypertension. Some studies also developed a method to prospect hypertension cases using machine learning, but not with the risk factor assessment based on the genotype data [1], [28].

Meanwhile, the use of deep learning models for magnetic resonance imaging (MRI) detection of brain tumors has shown promising results, but there is still much room for improvement. Deep learning models like convolutional neural networks (CNNs) are data-intensive and computationally expensive, requiring a large amount of brain tumor image data for training and high graphics processing unit (GPU) runtime and memory usage. To address these limitations, researchers have proposed methods such as ERV-Net, which reduces computational complexity but sacrifices some accuracy [28]. For successful and prompt treatment of brain cancers, early identification and classification are essential. However, conventional machine learning approaches have a hard time analysing the massive amounts of data generated by images. It remains a difficult and intricate problem, despite the fact that the well-known optimization technique stochastic gradient descent (SGD) has shown remarkable performance on large-scale assignments. Commonly used for hypertension classification, photoplethysmography has an accuracy of just 76% even when using deep learning. These findings stress the need for more study and development in the area of disease detection [29].

A particular kind of recurrent neural network (RNN) that is adept at seeing sequential patterns and linkages in the data is also used in the research of long short-term memory (LSTM) networks for forecasting stock market trends in Bangladesh. RNNs of the LSTM type are widely used in financial market forecasting because they are well-suited for the analysis of time-series data [30]. The minimal loss and average loss criteria are used to assess the LSTM model's performance. The study showed that the optimizer selection can have a significant impact on classification, while other parameters, including the number of LSTM layers, had little effect [31]. Thus, the goal of this work is to forecast the chance of getting hypertension using SNP data. Genetic variants known as SNPs are found at a particular location in a deoxyribonucleic acid (DNA) sequence. They offer important details regarding the inherited susceptibility to diseases like hypertension. In order to efficiently capture and reproduce long-term connections in sequential data, the study uses a RNN model called the LSTM model. Adaptive moment (Adam) is a deep learning model optimizer that considers gradients and learning rate simultaneously. The method is a cross between AdaGrad and RMSprop, and its performance depends on the nature of the particular issue at hand, the architecture of the neural network, and the computing power available. Based on the computed error levels, the optimizer iteratively modifies weights and parameters to improve model performance [32]. The study is unique in that it uses Adam optimization to integrate SNP data with LSTM, improving both overall performance and prediction accuracy. SNP data preparation and collecting are included in the design and development plan. The suggested method entails creating and implementing an LSTM model to forecast hypertension. The Adam optimizer is used to train the model, with an emphasis on optimizing hyperparameters. Metrics like area under the curve (AUC), sensitivity, specificity, and accuracy are used to assess the performance of the model. Furthermore, the LSTM model is contrasted with traditional machine learning techniques, such as decision tree (DT), K-nearest neighbor (KNN), Naïve Bayes (NB), and support vector machine (SVM).

2. METHOD

We will show facts pertaining to data sources during this session. We then present the proposed methods, including, among others, LSTM with Adam optimizer. The evaluation metric for this investigation is finally specified.

2.1. Data source

A new method for predicting the risk of hypertension uses information from SNPs. SNPs are differences in DNA sequences that influence only one nucleotide; these variations may help explain variances in hypertension and other genetic disorders. This study adopts a distinct approach by SNP data specifically, whereas other research may have placed greater emphasis on other kinds of data or prediction models. SNP data was utilized to predict the risk of hypertension using the OpenSNP platform. It is a website that allows users to share their genetic information, including SNPs. We retrieved SNP-containing genomic data from the OpenSNP database that were relevant to hypertension risk [33]. genome-wide association studies atlas data was used to compile the list of SNP and gene variants [34]. Based on data from the prodia laboratory, as shown in Table 1, 27 SNPs were chosen for their association with hypertension.

Table 1. The associated genes' genotypes

Gene	SNP code	Gene	SNP code
ACE	rs4341	MTHFR	rs17367504
ADD1	rs4961	NEDD4L	rs3865418
ADRB3	rs4994	NEDD4L	rs2288774
ATP2B1	rs17249754	PCSK1	rs6235
ATP2B1	rs2681472	PPARA	rs1800206
BGLAP	rs1800247	PPARA	rs4253778
CALCA	rs3781719	PRDM8-FGF5	rs11099098
CYP11B2	rs1799998	RNLS	rs2296545
CYP17A1	rs11191548	SMARCA4	rs1122608
DHFRP2	rs9266359	STK39	rs6749447
FGF5	rs1458038	TCEANC	rs2361159
FGF5	rs16998073	TNXB	rs2021783
KIF26B	rs10924160	ZFP64	rs6013382
LOC102723639-	rs35444		

2.2. The proposed method

Web scraping was one of several steps used to gather data for this study, which also included encoding the data into a numerical representation. After that, DTs, SVM, KNN, NB, and LSTM were among the categorization approaches used for modelling. Figure 1 shows the results of the subsequent evaluation of the performance levels. It would appear that this study makes use of a whole data analysis pipeline to find the best method for each task at hand, including data gathering, preprocessing, modelling with various machine learning algorithms, and thorough performance evaluation.

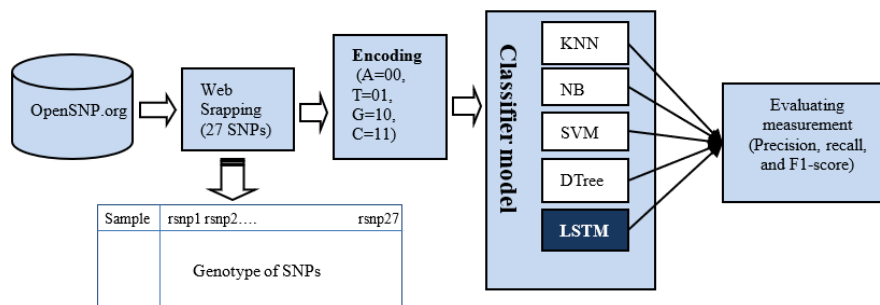


Figure 1. Block diagram of proposed method

2.3. Reproducibility

We used the QuerySNP package for data scraping in order to get a dataset from the OpenSNP website. After importing the dataset, the first thing that needs to be done is processing the data. This includes converting the data from DNA sequence characters (Adenine (A), Thymine (T), Guanine (G), and Cytosine (C)) to a binary number value by calculating the sum of each column/feature for each record of data. Then encoding the labels from text to numbers (0 and 1) for class. After that, the data needs to be divided into a train dataset and a test dataset; in this example, the percentage of data for the train set is 75%, while the piece of data for the test set is 25%. The train set will then be reformatted into an array of sequences to be used by LSTM. After then, the LSTM algorithm will be trained using the set that was previously utilized for training.

During the procedure (known as hyper-parameter tuning), early stopping and model checkpoint will be utilized to locate the most effective model. After the best model has been trained, the result of the best model's predictions will be assessed using confusion metrics once the training process has been completed.

For predicting using classifier model, we utilize TensorFlow library to build the LSTM-based model [35]. All hyper-parameters utilized in training the LSTM model with the Adam-optimizer will be explicitly stated and documented in the research paper and code repository. Detailed information on the training process, including data splitting, optimizer settings, learning rates, and batch sizes, will be provided for transparency. Then, we use the Sci-kit-learn library for implementing machine learning algorithm such as KNN, NB, SVM, and DT [36]. For reproducibility, the code repository will include detailed documentation, scripts, and instructions covering data preprocessing, model architecture, hyper-parameters, optimizer configurations, and training procedures, we share our source code in GitHub.

2.4. Web scrapping

In this step, we retrieve genotype data for 27 SNPs on opensnp.org which involves using automated techniques to extract genetic data from the opensnp.org website. We utilized QuerySNP library for web scrapping [33], which is a method used to automatically collect data from web pages to extract the desired genetic information and store it for analysis or further use. Analysing the web page structure, after we downloaded the web page, we analysed its HTML structure to find a way to extract the required SNP genotype data involving identifying HTML tags and desired attributes. After we extracted data, we stored it in an appropriate format in a comma-separated values (CSV) file. The details are stated in Algorithm 1.

Algorithm 1. Data collecting

```

Download genotype raw data from files in OpenSNP.org
for i to n sample data
    data[i] ← load(file[i])
    read and remove unimportant information (data [i])
    datsnp ← get data [i] by selected rsSNP
("rs4341", "rs4961", "rs4994", "rs17249754", "rs2681472",
 "rs1800247", "rs3781719", "rs1799998", "rs11191548", "rs9266359", "rs1458038", "rs1
6998073", "rs10924160", "rs35444", "rs17367504", "rs3865418", "rs2288774", "rs6235"
, "rs1800206", "rs4253778", "rs11099098", "rs2296545", "rs1122608", "rs6749447", "rs
2361159", "rs2021783", "rs6013382")
end
alldata ← Combine datsnp for all sample data
write in csv file (alldata)

```

2.5. Encoding SNPs data

To represent genetic information, we implemented binary encoding genotype data by transforming the selected SNPs. The encoding data of genotype by transforming A=00, T=01, G=10, and C=11 is a method used to represent genetic information in a binary format, where each of the four nucleotide bases (Adenine, Thymine, Guanine, and Cytosine) is assigned a unique binary code. This encoding simplifies the storage and processing of genetic data for various computational applications. The 27 genes contain nucleotide base pairs as controls to serve as features in the formation of the classifier model.

2.6. Classifier methods

A subclass of supervised learning techniques called classifier models is created expressly for prediction problems. Assigning data points to predetermined groups or classes is the aim of categorization. Classifier models are taught to spot trends and forecast the class labels of fresh, untainted data. For the algorithm to learn, training data is required. This study employed a variety of classifiers, including deep learning LSTM and traditional machine learning KNN, NB, SVM, and DT).

2.6.1. K-nearest neighbor classifier

The KNN method is a supervised machine learning with distanced based approach used to solve classification and regression [37]. It predicts using distance measures, majority voting, and averaging the goal values. It performs well in low-dimensional spaces and is straightforward for small datasets and online learning but is computationally intensive for big datasets.

2.6.2. Naïve Bayes classifier

Based on Bayes theorem, the NB algorithm is a probabilistic machine learning classification technique. The method determines a hypothesis' likelihood based on observed evidence, such as characteristics or traits. It makes the frequently incorrect assumption that the characteristics used for classification are conditionally independent. Probabilities are used by NB for categorization, prediction, and

training. It may be used to distributions that are multinomial, gaussian, or bernoulli. Simplicity, ease of use, high dimensionality, and performance even under the “naïve” independence assumption are some of its benefits. It might not be appropriate in real-world situations when the independence assumption is broken or if intricate interactions between characteristics must be represented. This method is implemented for clinical disease [38].

2.6.3. Support vector machine

For classification and regression issues, SVM, a supervised machine learning technique, is used. It seeks to maximize the margin between classes while identifying the hyperplane that optimally divides data points into distinct classes. The method starts with a labelled dataset, and to enhance performance, features are scaled or normalized. The kernel and the regularization parameter (C) are the two primary hyper-parameters selected. The regularization parameter regulates the trade-off between margin and classification mistakes, while the kernel function specifies how data is transformed into a higher-dimensional space. By resolving a convex optimization problem and determining the best hyperplane, SVM ensures that all of the data points are accurately categorised. With the right regularization choice, it is efficient in high-dimensional spaces, adaptable, and resistant to over-fitting. Although it does not directly produce probability estimates, it can be computationally costly and sensitive to the choice of kernel and hyper-parameters [39].

2.6.4. Decision tree

To solve classification and regression issues, machine learning techniques known as DTs are used. The root is at the top, while the leaves are at the bottom, making up its nodes. The method separates the data into subsets according to the values of the input characteristics, and each leaf of the tree represents a prediction or class label. The depth of the tree is defined as the number of edges between the root node and the deepest leaf. To maximize information gain or reduce mean squared error, the algorithm splits points at each node and chooses the optimal feature during training. Criteria like gini impurity, entropy, or mean squared error are used to choose the feature. When specific circumstances are satisfied, the tree-growing process comes to the end of nodes (leaves) [40].

2.6.5. Long short-term memory

A RNN architecture called LSTM was created to identify long-term relationships and patterns in sequential input. It fixes the issue with standard RNNs’ vanishing gradients, which makes it challenging to recognize and recall long-range relationships. The cell state (C_t), a continuous vector that makes up the LSTM cells, can be read, reset, or updated using a series of gates. In addition to the cell state, the hidden state (h_t) is a continuous vector that is utilized to transmit information across time steps. In LSTM cells, you will find three distinct kinds of gates: input, forget, and output. The concealed and cell states are each modulated by a specific gate. Each gate in a cell state computation has a specific purpose: the input gate adds new data, the forget gate removes data, and the output gate shows or hides the hidden data. Deep learning is a more advanced kind of artificial neural networks that makes use of numerous hidden layers. As illustrated in Figure 2, this study suggests utilizing three LSTMs with concealed layers. This approach is used when other machine learning methods fail to resolve complicated situations. To address the vanishing gradient issue that arises during back-propagation, the LSTM was developed [41], [42]. Long-term dependencies are handled by LSTM using memory cells, which also decide whether to keep or delete the information. Three gates on each memory cell control the information flow. The input gate determines what should be added to the cell, the output gate creates new long-term memory, and the forget gate selects what should be removed from the previous memory unit. The LSTM’s input can be states as $x=(x_1, x_2, \dots, x_t)$, and the output as $y=(y_1, y_2, \dots, y_t)$ [42]. The methods described below can be used to compute its output (1), disregard gate f_t .

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

Remark:

f_t : forget gate

σ : function of sigmoid

W_f : weight of forget gate

h_{t-1} : hidden state in the previous data

x_t : input from LSTM

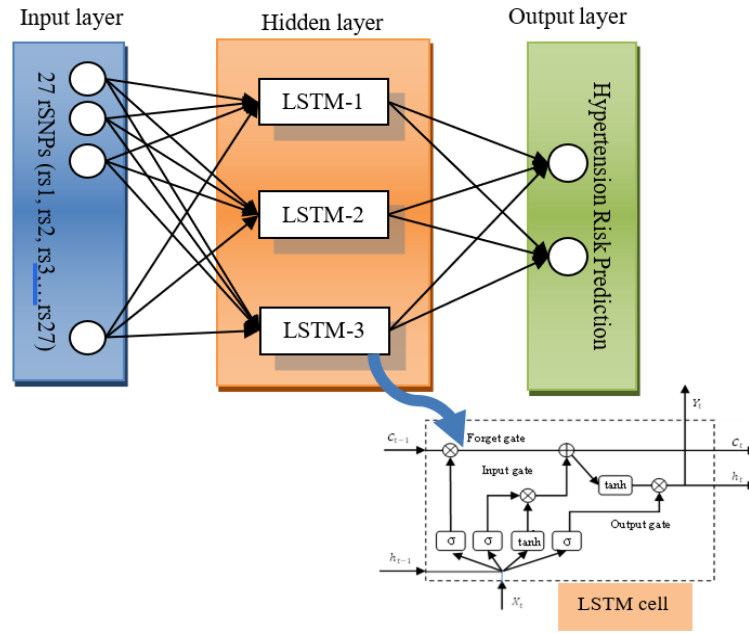


Figure 2. The architecture of the proposed method

The second step is to compute the input gate i_t using (2) and (3):

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\hat{c}_t = \tanh(W_c * [h_{t-1}, x_t] + b_c) \tag{3}$$

- i_t : input gate
- W_i : weight input gate
- b_i : bias input gate
- \hat{c}_t : candidate cell state
- \tanh : hyperbolic tangent function
- W_c : weight candidate cell state
- b_c : bias candidate cell state

then, the third step is to update the cell state using (4):

$$C_t = f_t * C_{t-1} + i_t * \hat{c}_t \tag{4}$$

- C_t : cell state
- C_{t-1} : cell state value of the previous data

the fourth step is to compute the output gate using (5) and (6):

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t * \tanh(c_t) \tag{6}$$

- o_t : output gate
- W_o : weight output gate
- b_o : bias output gate
- h_t : hidden state

Finally, the stage is to calculate the predicted value (y_t) as stated in (7):

$$\hat{y}_L = W_y h_t + b_y \tag{7}$$

- \hat{y}_L : the predicted value or the output of LSTM
- W_y : weight of layer output

b_y : bias of layer output
the function of sigmoid $\sigma(x)$ is shown in (8):

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (8)$$

the function of hyperbolic tangent, $\tanh(x)$ is stated in (9).

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (9)$$

The LSTM model used in this research is modified to accommodate the data so the model can produce better results. The modified LSTM consists of three LSTMs (stacked LSTMs) with a drop-out layer after each LSTM layer to construct the model is not over-fitted. We applied a sigmoid dense activation layer, binary cross entropy (BCE) loss function for binary classification, and Adam as its optimization algorithm.

2.7. Adam optimizer

Based on the history of gradients, AdaGrad is an adaptive optimizer that modifies the learning rate for each weight. The learning rate is divided by the running total of the gradients' previous squares. which has the effect of reducing the learning rate for frequently occurring weights. Adam is a hybrid that incorporates both RMSprop and AdaGrad. By modifying the initial and final moments of the gradients, it takes into account the learning rate for every weight. Because of its effectiveness and user-friendliness, it ranks among the most used optimizers in deep learning. The problem type, neural network design, and available computing resources all play a role in selecting an optimizer method. Finding the optimal optimizer for a certain problem often requires some trial and error with several tools. Finding the global minimum value of the convergence of the loss function is possible with the help of the optimizer in a neural network. To a large extent, the optimizer contributes to the improvement of the model's correctness by decreasing the error value with each cycle [43]. Taking the actual value and dividing it by the projected value gives the model's error value. The collected error values will be used to update the model's parameters and weights. These parameters and weights are updated using the back-propagation process.

Several flexible measuring techniques exist, including the Adam method. A stochastic optimization algorithm with a first-order gradient in memory and a high efficiency is Adam. The optimization algorithm Adam was developed by using two optimization algorithms, AdaGrad, sparse gradients and RMSprop, which have superior performance when analysing non-stationary data. To update weight of Adam's optimizer will be available in (10) and (11), respectively [44].

$$vt = \beta_1 * v_{t-1} - (1 - \beta_1) * gt \quad (10)$$

$$st = \beta_2 * s_{t-1} - (1 - \beta_2) * gt_2 \quad (11)$$

$$w_{new} = w_{old} - \alpha vst + * gt \quad (12)$$

vt : gradient's exponential moving average

st : square gradient moving at an exponential rate hyper-parameter in RMSprop

gt : gradient at the t^{th} time-step

w_{old} : weight of the previous learning rate

2.8. Binary cross entropy function

The deep learning method requires the LSTM algorithm to regulate the error of the classifier model. BCE is used as a loss function to continuously monitor the condition. The weights are updated using the function to decrease the error rate and optimize the classifier model during the training phase. The BCE class encodes the output into two binary classes, "hypertension risk" or "normal," which are the target classes used in this study. Using BCE on each output individually does not necessarily mean that each state may fall under more than one class. Instead, it means that each output is treated as a separate binary classifier problem, where the class value is either 0 or 1 for each output. This approach is often used in multi-label classification problems, where a single input may belong to multiple classes. In this case, the model outputs a probability distribution over all possible classes for each input, and the BCE loss is calculated independently for each output.

To calculate the BCE loss for a multi-label classification problem, we first convert the true labels into a binary matrix, where each column is a potential class, and each row represents an input. If a class

applies to the input, the appropriate matrix entry is set to 1; otherwise, it is set to 0. We then compare this binary matrix with the predicted probability distribution using the BCE loss function, which is calculated independently for each output [45]. The BCE calculates the discrepancy between the class 1 prediction's true and projected probability distributions. The BCE is defined in (13):

$$BCE = -(y \times \log(p)) + (1 - y) \times \log(1 - p) \quad (13)$$

where y is the true label (either 0 or 1), p is the predicted probability for class 1, and \log is the natural logarithm. When the predicted probability is close to the true label (either 0 or 1), the BCE value will be close to zero, indicating a better model performance. The BCE is used as a measure of error to be minimized during the training of the neural network.

2.9. Performance evaluation

In addition to precision, recall, and F1-score, accuracy is another common metric used to evaluate the performance of a classification model. Here's how to calculate accuracy, precision, recall, and F1-score for hypertension risk prediction. The percentage of correctly categorized persons in the dataset (i.e., the sum of true positives and true negatives) is known as accuracy in (14).

$$Accuracy = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (14)$$

Where FN, FP, TN, TP, and FN represent the number of false negatives, false positives, true negatives, false negative, and respectively. False positives are thought to be at high risk for developing hypertension but do not actually do so (individuals who are predicted to be at low risk for hypertension). Precision is the percentage of people who are correctly identified as being at high risk for developing hypertension out of all those who are projected to be at high risk as shown in (15).

$$Precision = \frac{TP}{(TP+FP)} \quad (15)$$

The next step is to determine the percentage of results that are accurate positives. They are anticipated to have high blood pressure. The practice of making this prediction among all individuals who already have high blood pressure is called recall in (16).

$$Recall = \frac{TP}{(TP+FN)} \quad (16)$$

The F1-score is the harmonic means of accuracy and recall and gives a balanced assessment of the two metrics. It is derived by dividing precision by recall. In (17) expresses the measurement in general prediction:

$$F1 - score = 2 \times \frac{(precision \times recall)}{(precision+recall)} \quad (17)$$

these metrics are evaluated together to comprehensively know well the hypertension risk prediction model performs. Different thresholds for risk prediction may lead to different values of these metrics. The choice of threshold should be based on the specific application and goals of the model.

3. RESULTS AND DISCUSSION

The performance measures (recall, precision, and F1-score) are evaluated to represent machine learning, including KNN, NB, SVM, DT, and the modified LSTM. The performance results for all algorithms are shown in Figure 3. In Figure 3, the SVM has a high recall but a low accuracy and F1-score. This algorithm tries to make the space between classes as big as possible so that there is a clear line between them. In some cases, trying to get a clear range could cause more false positives (FP), which would make the accuracy worse. High recall means that the model is good at catching a lot of real positive cases, but it may do this by including a lot of fake positives as well. Because of this, the precision goes down, which influences the F1-score because it takes both accuracy and recall into account. It may cause of imbalanced dataset distribution. In general, LSTM is high performance in all measurements.

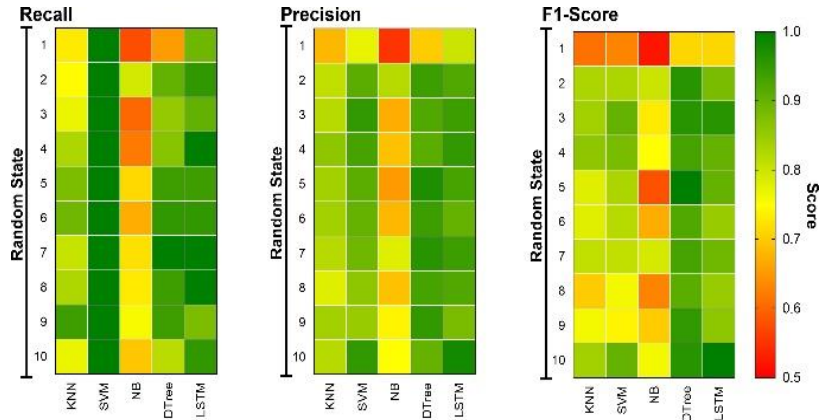


Figure 3. Performance comparison of KNN, SVM, NB, DT, and LSTM

Then, when compared for all algorithms, the accuracy rate of LSTM is dominant to the other algorithms. In general, the accuracy of the proposed method was achieved more than 0.8 in all random states. The NB algorithm achieved the lowest accuracy rate, almost less than 0.7 in all random states, as shown in Figure 4. Furthermore, the low performance rate is caused by misclassification in determining the actual class. It can be found by calculating FP and false negative (FN). The higher both values, the lower the performance. The proposed method, LSTM, is stable and low for the FP and FN values as shown in Figures 5 and 6, respectively.

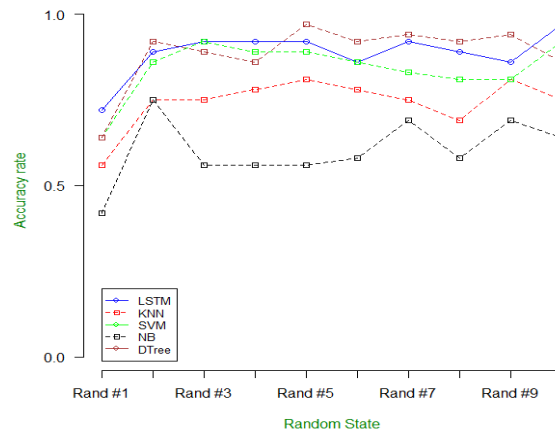


Figure 4. Accuracy comparison of LSTM to the conventional machine learning algorithm

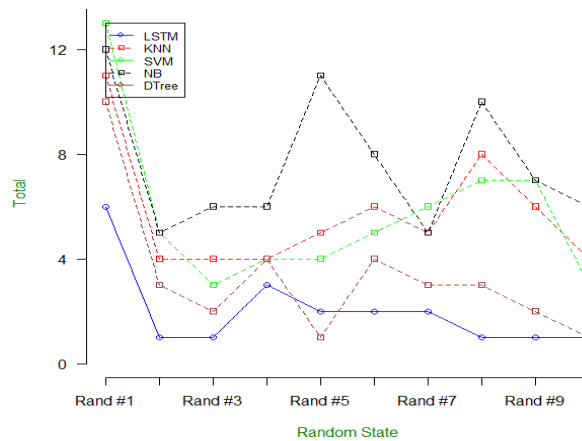


Figure 5. The FP comparison of LSTM to machine learning algorithm

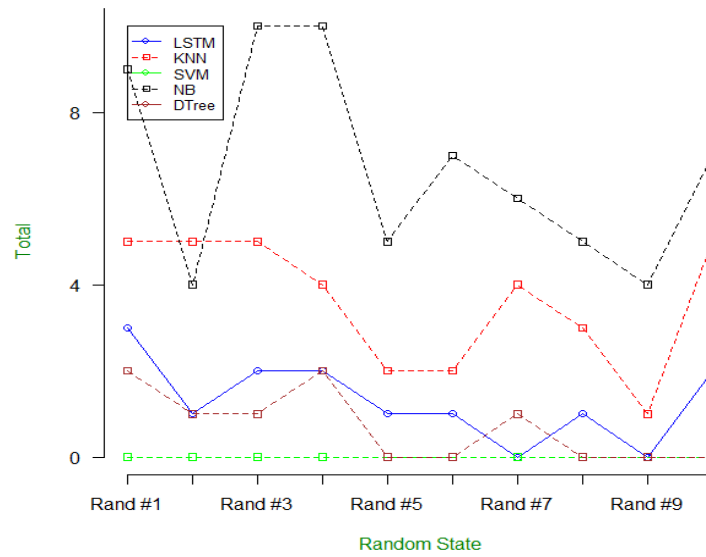


Figure 6. The FN comparison of LSTM to machine learning algorithm

As can be seen in Figure 7, the training data set for the LSTM approach was generated as a decent classifier model at an epoch that was higher than the 350th. When it got to this point, the performance result was good. It indicates that the approach can make accurate predictions.

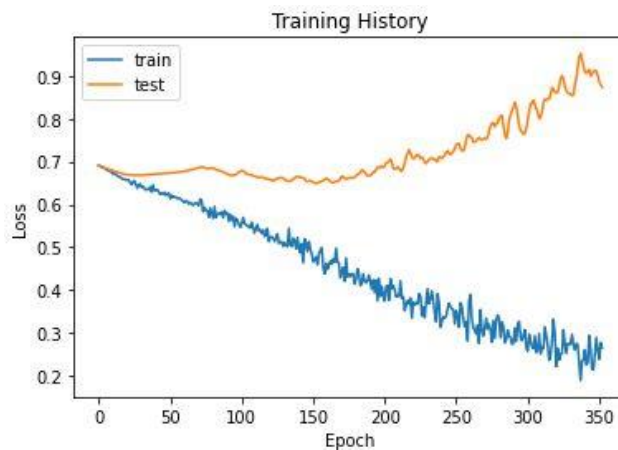


Figure 7. The loss function of LSTM

However, the weakness of the modified LSTM method is computational time as shown in Figure 8. It required high computation to construct a classifier model using training data compared to the other algorithms as illustrated in Figure 8(a). In the first random state, the computational time of the proposed method is required until 35.24 s as required in training data, even though the next state is required to gradually reduce computational time until below 5 s as shown in Figures 8(b).

Hypertension is a disease that has a high prevalence in the world causing death. This disease also affects the performance of other organs including the cardiovascular system. Various attempts have been made to treat this disease, ranging from the classical method based on blood pressure or the electrocardiogram (ECG) and a photoplethysmography (PPG) signals measurement from patients [27]. However, both types of measurement are healing efforts. Furthermore, many studies on genomic as one of hypertension risk factors [22]. Therefore, this study proposes a method of preventing hypertension through early prediction based on a genomic approach through differences of SNP. Research on hypertension prediction using the whole SNPs genome using a machine learning method needs to be implemented as a feature selection method [1]. Since the limited SNP data provided without any information, which one the

significant feature, then this study proposed, the deep learning method, LSTM algorithm. Using several related SNPs is proposed as a data set for the learning process to construct an LSTM classification model with a BCE. The loss functions are used to determine the amount that a model should try to minimize during training using 32 units of hidden layers, and 0.2 drop-out rate. This function is used in relation to determine two types of output variable values, namely at risk of hypertension or not (normal) [27].

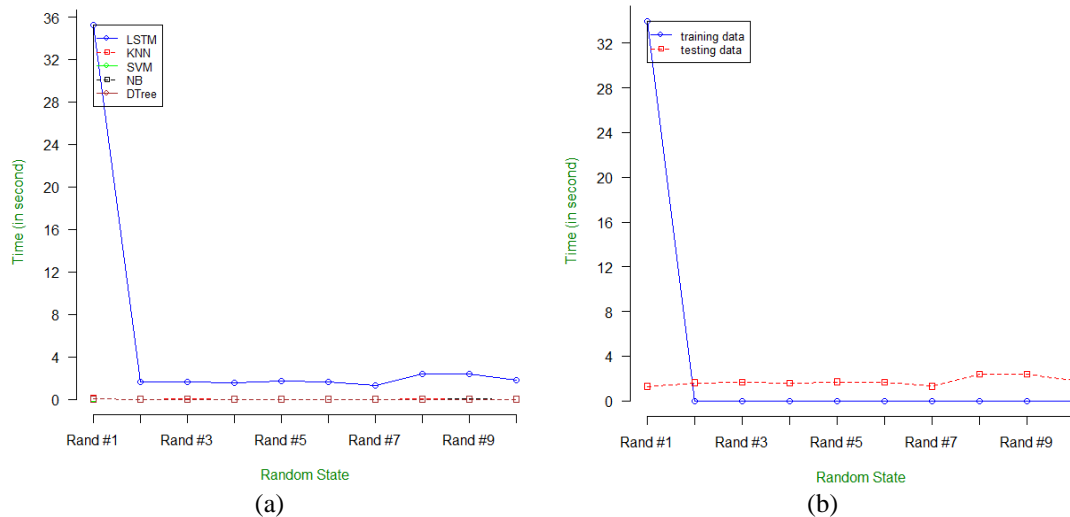


Figure 8. Computational time comparison of (a) LSTM to machine learning algorithm and (b) training and testing data in LSTM

The LSTM model with Adam optimizer achieved a significantly better result compared to sigmoid SGD optimizer and the other machine learning methods such as NB, KNN, SVM, and DT. The LSTM's result is stable across multiple epochs/random state compared to other methods that have high accuracy in one random state and very low accuracy in other random state. Based on the experimental results, the proposed method has a stable high-performance value compared to classical machine learning methods as shown in Table 2. The performance of recall, precision, accuracy, and F1-score on average of ten times examination randomly are 0.96, 0.88, 0.89, and 0.92 respectively. In general, the LSTM with Adam optimizer achieved is dominant for result performance.

Table 2. Performance of evaluation result

Algorithm	Recall	Precision	Accuracy	F1-score
KNN	0.85	0.78	0.74	0.81
SVM	1.00	0.81	0.84	0.89
NB	0.72	0.69	0.60	0.70
DT	0.90	0.92	0.89	0.91
LSTM-Adam optimizer	0.96	0.88	0.89	0.92
LSTM-SGD optimizer	1.00	0.66	0.66	0.79

4. CONCLUSION

Predicting hypertension risk using LSTM method with Adam optimizer achieved high performance, especially for recall, and F1-score, on average of 0.96 and 0.92 respectively. The NB has the lowest performance. However, for modelling LSTM classification is required high computational time. Therefore, it is necessary to develop the tuning parameter and topology of network in LSTM to reduce the epoch of computational time during the classifier model using training data.

ACKNOWLEDGEMENT

The superior grant program “*Hibah Penelitian Unggulan (HPU) 2022*” of Brawijaya University, under contract number 975.22/UN10.C10/PN/2021, is funding this study.





REFERENCES

- [1] C. Ye *et al.*, "Prediction of incident hypertension within the next year: prospective study using statewide electronic health records and machine learning," *Journal of Medical Internet Research*, vol. 20, no. 1, p. e22, Jan. 2018, doi: 10.2196/jmir.9268.
- [2] C. Maj *et al.*, "Dissecting the polygenic basis of primary hypertension: identification of key pathway-specific components," *Frontiers in Cardiovascular Medicine*, vol. 9, Feb. 2022, doi: 10.3389/fcvm.2022.814502.
- [3] C. Armstrong, "JNC 8 guidelines for the management of hypertension in adults," *American Family Physician*, vol. 90, no. 7, pp. 503–504, 2014.
- [4] J. Skeete and D. J. DiPette, "Genetics of hypertension: implications of single nucleotide polymorphism(s) in African populations and beyond," *Journal of Clinical Hypertension*, vol. 20, no. 3, pp. 496–498, Mar. 2018, doi: 10.1111/jch.13249.
- [5] M. Íñiguez *et al.*, "ACE gene variants rise the risk of severe COVID-19 in patients with hypertension, dyslipidemia or diabetes: a spanish pilot study," *Frontiers in Endocrinology*, vol. 12, Aug. 2021, doi: 10.3389/fendo.2021.688071.
- [6] H. Jin, Y. Huang, and G. Yang, "Association between α -adducin rs4961 polymorphism and hypertension: a meta-analysis based on 40 432 subjects," *Journal of Cellular Biochemistry*, vol. 120, no. 3, pp. 4613–4619, Mar. 2019, doi: 10.1002/jcb.27749.
- [7] L. Wang *et al.*, "Association between single-nucleotide polymorphisms in six hypertensive candidate genes and hypertension among northern Han Chinese individuals," *Hypertension Research*, vol. 37, no. 12, pp. 1068–1074, Dec. 2014, doi: 10.1038/hr.2014.124.
- [8] M. Xie *et al.*, "ATP2B1 gene polymorphisms rs2681472 and rs17249754 are associated with susceptibility to hypertension and blood pressure levels: a systematic review and meta-analysis," *Medicine (United States)*, vol. 100, no. 15, p. E25530, Apr. 2021, doi: 10.1097/MD.00000000000025530.
- [9] Y. Ling, X. Gao, H. Lin, H. Ma, B. Pan, and J. Gao, "A common polymorphism rs1800247 in osteocalcin gene is associated with hypertension and diastolic blood pressure levels: The Shanghai Changfeng study," *Journal of Human Hypertension*, 2016.
- [10] L. Sydorchuk *et al.*, "The cytochrome 11B2 aldosterone synthase gene rs1799998 single nucleotide polymorphism determines elevated aldosterone, higher blood pressure, and reduced glomerular filtration, especially in diabetic female patients," *Endocrine Regulations*, vol. 54, no. 3, pp. 217–226, Jul. 2020, doi: 10.2478/enr-2020-0024.
- [11] X. Lu *et al.*, "Genetic predisposition to higher blood pressure increases risk of incident hypertension and cardiovascular diseases in Chinese," *Hypertension*, vol. 66, no. 4, pp. 786–792, Oct. 2015, doi: 10.1161/HYPERTENSIONAHA.115.05961.
- [12] F. E. R. Punzalan *et al.*, "The rs1458038 variant near FGF5 is associated with poor response to calcium channel blockers among Filipinos," *Medicine (United States)*, vol. 101, no. 5, p. e28703, Feb. 2022, doi: 10.1097/MD.00000000000028703.
- [13] H. Jeong, H. S. Jin, S. S. Kim, and D. Shin, "Identifying interactions between dietary sodium, potassium, sodium–potassium ratios, and fgf5 rs16998073 variants and their associated risk for hypertension in korean adults," *Nutrients*, vol. 12, no. 7, pp. 1–14, 2020, doi: 10.3390/nu12072121.
- [14] Y. M. Park, C. K. Kwock, K. Kim, J. Kim, and Y. J. Yang, "Interaction between single nucleotide polymorphism and urinary sodium, potassium, and sodium-potassium ratio on the risk of hypertension in Korean adults," *Nutrients*, vol. 9, no. 3, p. 235, Mar. 2017, doi: 10.3390/nu9030235.
- [15] Y. Wang and J.-G. Wang, "Genome-wide association studies of hypertension and several other cardiovascular diseases," *Pulse*, pp. 169–186, 2019.
- [16] B. Xi, Y. Shen, K. H. Reilly, X. Wang, and J. Mi, "Recapitulation of four hypertension susceptibility genes (CSK, CYP17A1, MTHFR, and FGF5) in East Asians," *Metabolism: Clinical and Experimental*, vol. 62, no. 2, pp. 196–203, Feb. 2013, doi: 10.1016/j.metabol.2012.07.008.
- [17] A. Maciejewska-Skrendo *et al.*, "Does the PPARA intron 7 gene variant (rs4253778) influence performance in power/strength-oriented athletes? a case-control replication study in three cohorts of european gymnasts," *Journal of Human Kinetics*, vol. 79, no. 1, pp. 77–85, Jul. 2021, doi: 10.2478/hukin-2020-0060.
- [18] L. L. Zhong *et al.*, "Systolic hypertension related single nucleotide polymorphism is associated with susceptibility of ischemic stroke," *European review for medical and pharmacological sciences*, vol. 21, no. 12, pp. 2901–2906, 2017.
- [19] Q. F. Chen *et al.*, "Correlation of rs1122608 SNP with acute myocardial infarction susceptibility and clinical characteristics in a Chinese Han population: a case-control study," *Anatolian Journal of Cardiology*, vol. 19, no. 4, pp. 249–258, 2018, doi: 10.14744/AnatolJCardiol.2018.35002.
- [20] T. Kunnas, K. Määttä, S. T. Nikkari, and Y. Wang, "Variant rs6749447 (T>G) in the serine threonine kinase gene is associated with cardiovascular complications, the Tampere adult population cardiovascular risk study," *Medicine (United States)*, vol. 100, no. 42, p. E27566, Oct. 2021, doi: 10.1097/MD.00000000000027566.
- [21] Y. Hiura *et al.*, "A genome-wide association study of hypertension-related phenotypes in a Japanese population," *Circulation Journal*, vol. 74, no. 11, pp. 2353–2359, 2010, doi: 10.1253/circj.CJ-10-0353.
- [22] R. Alzubi, N. Ramzan, H. Alzoubi, and S. Katsigiannis, "SNPs-based hypertension disease detection via machine learning techniques," in *ICAC 2018 - 2018 24th IEEE International Conference on Automation and Computing: Improving Productivity through Automation and Computing*, Sep. 2018, pp. 1–6, doi: 10.23919/ICAC.2018.8748972.
- [23] R. Dash *et al.*, "Computational SNP analysis and molecular simulation revealed the most deleterious missense variants in the NBD1 domain of human ABCA1 transporter," *International Journal of Molecular Sciences*, vol. 21, no. 20, pp. 1–23, Oct. 2020, doi: 10.3390/ijms21207606.
- [24] S. J. Wu *et al.*, "Particle swarm optimization algorithm for analyzing SNP-SNP interaction of renin-angiotensin system genes against hypertension," *Molecular Biology Reports*, vol. 40, no. 7, pp. 4227–4233, Jul. 2013, doi: 10.1007/s11033-013-2504-8.
- [25] L. Muflikhah, N. Hidayat, and D. J. Hariyanto, "Prediction of hypertension drug therapy response using K-NN imputation and SVM algorithm," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 15, no. 1, pp. 460–467, Jul. 2019, doi: 10.11591/ijeecs.v15.i1.pp460-467.
- [26] Y. Shaikh, V. Parvati, and S. R. Biradar, "Early disease prediction algorithm for hypertension-based diseases using data aware algorithms," *Indonesian Journal of Electrical Engineering and Computer Science (IJECS)*, vol. 27, no. 2, pp. 1100–1108, Aug. 2022, doi: 10.11591/ijeecs.v27.i2.pp1100-1108.
- [27] E. Martinez-Ríos, L. Montesinos, M. Alfaro-Ponce, and L. Pecchia, "A review of machine learning in hypertension detection and blood pressure estimation based on clinical and physiological data," *Biomedical Signal Processing and Control*, vol. 68, p. 102813, Jul. 2021, doi: 10.1016/j.bspc.2021.102813.
- [28] Z. Y. Luo *et al.*, "A study of machine-learning classifiers for hypertension based on radial pulse wave," *BioMed Research International*, vol. 2018, pp. 1–12, Nov. 2018, doi: 10.1155/2018/2964816.
- [29] C. T. Yen, S. N. Chang, and C. H. Liao, "Deep learning algorithm evaluation of hypertension classification in less photoplethysmography signals conditions," *Measurement and Control (United Kingdom)*, vol. 54, no. 3–4, pp. 439–445, Mar. 2021, doi: 10.1177/00202940211001904.





- [30] M. A. Islam, M. R. Sikder, S. M. Ishtiaq, and A. Sattar, "Stock market prediction of Bangladesh using multivariate long short-term memory with sentiment identification," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 5, pp. 5696–5706, 2023, doi: 10.11591/ijece.v13i5.pp5696-5706.
- [31] M. Sher, N. Minallah, T. Ahmad, and W. Khan, "Hyperparameters analysis of long short-term memory architecture for crop classification," *International Journal of Electrical and Computer Engineering*, vol. 13, no. 4, pp. 4661–4670, 2023, doi: 10.11591/ijece.v13i4.pp4661-4670.
- [32] S. P. K. Gudla and S. K. Bhoi, "A study on effect of learning rates using adam optimizer in LSTM deep intelligent model for detection of DDoS attack to support fog based IoT systems," in *Communications in Computer and Information Science*, vol. 1729 CCIS, 2022, pp. 27–38.
- [33] B. Greshake, P. E. Bayer, H. Rausch, and J. Reda, "openSNP-A crowdsourced web resource for personal genomics," *PLoS ONE*, vol. 9, no. 3, p. e89204, Mar. 2014, doi: 10.1371/journal.pone.0089204.
- [34] K. Watanabe *et al.*, "A global overview of pleiotropy and genetic architecture in complex traits," *Nature Genetics*, vol. 51, no. 9, pp. 1339–1348, 2019, doi: 10.1038/s41588-019-0481-0.
- [35] M. Abadi *et al.*, "TensorFlow: large-scale machine learning on heterogeneous distributed systems," *arXiv e-prints*, 2016, doi: 10.48550/arXiv.1603.04467.
- [36] F. Pedregosa *et al.*, "Scikit-learn: machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [37] N. Ali, D. Neagu, and P. Trundle, "Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets," *SN Applied Sciences*, vol. 1, no. 12, p. 1559, Dec. 2019, doi: 10.1007/s42452-019-1356-9.
- [38] V. Jackins, S. Vimal, M. Kaliappan, and M. Y. Lee, "AI-based smart prediction of clinical disease using random forest classifier and Naive Bayes," *Journal of Supercomputing*, vol. 77, no. 5, pp. 5198–5219, May 2021, doi: 10.1007/s11227-020-03481-x.
- [39] M. Awad and R. Khanna, "Support vector machines for classification," in *Efficient Learning Machines*, Berkeley, CA: Apress, 2015, pp. 39–66.
- [40] I. D. Mienye, Y. Sun, and Z. Wang, "Prediction performance of improved decision tree-based algorithms: a review," *Procedia Manufacturing*, vol. 35, pp. 698–703, 2019, doi: 10.1016/j.promfg.2019.06.011.
- [41] A. Kulshrestha, V. Krishnaswamy, and M. Sharma, "Bayesian BiLSTM approach for tourism demand forecasting," *Annals of Tourism Research*, vol. 83, 2020, doi: 10.1016/j.annals.2020.102925.
- [42] K. E. ArunKumar, D. V. Kalaga, C. M. S. Kumar, M. Kawaji, and T. M. Brenza, "Forecasting of COVID-19 using deep layer recurrent neural networks (RNNs) with gated recurrent units (GRUs) and long short-term memory (LSTM) cells," *Chaos, Solitons and Fractals*, vol. 146, p. 110861, May 2021, doi: 10.1016/j.chaos.2021.110861.
- [43] H. Jindal, N. Sardana, and R. Mehta, "Analyzing performance of deep learning techniques for Web navigation prediction," *Procedia Computer Science*, vol. 167, pp. 1739–1748, 2020, doi: 10.1016/j.procs.2020.03.384.
- [44] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.
- [45] S. Saxena, "Binary cross entropy/log loss for binary classification," *AnalyticsVidhya.com*, 2021. <https://www.analyticsvidhya.com/blog/2021/03/binary-cross-entropy-log-loss-for-binary-classification/> (accessed Dec. 01, 2022).

BIOGRAPHIES OF AUTHORS






Lailil Muflikhah     received her B.Sc. in computer science from the Institute of Technology Sepuluh Nopember, his M.Sc. in computer science from Universiti Technology of Petronas in Malaysia, and his Ph.D. in bioinformatics from the University of Brawijaya. She is presently an Associate Professor at the University of Brawijaya's Faculty of Computer Science's Department of Informatics Engineering. Her areas of interest in study are artificial intelligence, machine learning, and bioinformatics. She can be contacted at email: lailil@ub.ac.id.






Nashi Widodo     is a highly accomplished researcher and professor with a diverse educational background. Having earned a bachelor's degree from Brawijaya University, a master's degree from the Bandung Institute of Technology, and a PhD from the University of Tsukuba in Japan, he has cultivated a rich academic foundation. Achieving full professorship in cancer biology in 2018, he has become a recognized authority in the field. His current focus revolves around pioneering research in natural products for cancer treatment, contributing significantly to advancements in this critical area of medical science. He can be contacted at email: widodo@ub.ac.id.






Imam Cholissodin    obtained his master's degree in information engineering from FTIF ITS Surabaya. He has been working as a lecturer at the University of Brawijaya, Malang Faculty of Computer Science since 2012. As part of the Intelligent Laboratory Computation and research between BRIN x UB, he is currently enrolled in a doctoral program in computer science at ITS with a focus on developing core engine AI, ML, and deep learning for the bio-molecular field. This engine will be integrated with Big Data App as a general library for any programming language and OS platforms. In order to establish and develop Advanced Technology "Smart App" products in the Industrial Revolution 4.0 and Society 5.0 eras, his research focuses on backend and frontend computation on any locally based, serverless, and health, governance, and other devices. He can be contacted at email: imamcs@ub.ac.id.






Feri Eko Hermanto    received his B.Sc. from the Biology Department at the University of Brawijaya in 2018, and via the Master-lead-to-Ph.D. program run by the Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia, he earned his Ph.D. from the same university in 2022. His present research focuses on animal physiology and the bioprospecting of animal products for human health at the University of Brawijaya's Faculty of Animal Sciences. He can be contacted at email: fe.hermanto@ub.ac.id.






Teresa Liliana Wargasetia    is a professor in molecular biology at Faculty of Medicine, Universitas Kristen Maranatha, Bandung, Indonesia. She graduated from Indonesia's Institut Teknologi Bandung with a bachelor's degree in biology. She received master and doctoral degrees in medical science from Universitas Padjadjaran, Bandung, Indonesia. Her area of interest in study is molecular pathobiology. She can be contacted at email: Teresa.lw@med.maranatha.edu.



Dr. Hana Ratnawati, dr., M. Kes.,    is a lecturer at Maranatha Christian University, Indonesia, in the Department of Histology. She has a master's degree in Pathobiology from Padjadjaran University, Indonesia, and has conducted research on cancer trophoblast. In 2006, she attended an immunology course in Amsterdam and received a scholarship to study at Vrije University. Her doctoral degree is from Gadjah Mada University, Yogyakarta, focusing on cancer immunology. She received a sandwich scholarship for dissertation research at Vrije University, Amsterdam, funded by the High Education Ministry of Indonesia. Her research interests include herbal medicine for treating cardiovascular diseases and other health issues. She can be contacted at email: hana.ratnawati@maranatha.ac.id.



Prof. Riyanarto Sarno    teaches at the Institut Teknologi Sepuluh Nopember in Surabaya, Indonesia. In 1987, he graduated from Indonesia's Institut Teknologi Bandung with a bachelor's degree in electrical engineering. In 1988 and 1992, respectively, he earned his M.Sc. and Ph.D. in computer science from the University of Brunswick in Canada. He received a promotion to full professor in 2003. IoT, process-aware information systems, intelligent systems, and smart grid are some of his areas of interest in both teaching and research. He can be contacted at email: riyanarto@if.its.ac.id.