International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 14, No.1 (Sep-2023)

http://dx.doi.org/10.12785/ijcds/140158

# Autoencoder-Based Feature Learning for Predicting Cardiovascular Disease

**Angelina Puput Giovani[1], Hilman Ferdinandus Pardede[1,2] and Agus Subekti[1,2]**

[1]*Graduate School of Computer Science, Nusa Mandiri University, Jakarta, Indonesia*
[2]*National Research and Innovation Agency, Jakarta, Indonesia*

*Received 27 Sep. 2022, Revised 08 May 2023, Accepted 31 Jul. 2023, Published 01 Sep. 2023*

**Abstract:** Cardiovascular disease is the world's leading cause of death. Some studies have used the machine learning method to predict cardiovascular diseases based on medical records. However, due to high correlation between data in medical records, much needs to be done in the field. Here, we propose to use Autoencoder based feature learning to predict cardiovascular disease, because Autoencoder can process complex, high-dimensional datasets by doing linear and non-linear projections. Thus, we hope the autoencoder can learn about non-linear and complex connections between the medical data being used. We varied the depth of autoencoder in this paper from 3 to 7 layers, and the depth of layer was varied to several neurons at the bottleneck. The results are then input into other classifiers, such as Logistic Regression, Naive Bayes, SVM, KNN, Decision Tree, XGBoost, Random Forest, and Neural Networks. Our experiments show that use of autoencoder-based feature learning can improves the performance of the classifier by 0.75%. However, we see that the depth of the layer does not always enhance performance and needs to be defined empirically.

**Keywords:** Cardiovascular, Predicting, Autoencoder, Deep Learning, Unsupervised Learning, Feature Learning, Classifier model

## 1. INTRODUCTION

The leading cause of death in Indonesia is cardiovascular disease. Based on data from the Basic Health Research (Riskesdas) in 2018 [1] the Coordinating Minister for Human Development and Culture (Menko PMK) revealed that the prevalence of heart disease in Indonesia is 10.5 percent, or 15 out of every 1,000 people, and that the disease is on the rise year over year. Several heart problems can lead to critical health problems and can even lead to sudden death. Cardiovascular disease risk factors include a variety of things like diabetes, high blood pressure, elevated stress, cholesterol, obesity, age, gender, smoking and drinking habits, an unhealthy diet, insufficient exercise, and family history. The increasing number of deaths from cardiovascular causes is due to the fact that it is impossible for a person to undergo medical tests such as an electrocardiogram (ECG) continuously [2],the need for sophisticated equipment for treatment and ignorance of cardiovascular symptoms due to expensive treatment [3].

The contribution of technology to preventing cardiovascular disease is expanding. The study of predicting cardiovascular disease using machine learning methods has been going on for decades. The use of machine learning is done by utilizing historical data or the medical records. Several machine learning algorithms have been carried out in predicting cardiovascular disease. Neural networks [4], decision tree [3]have been employed in previous studies. In one study [5], Naive Bayes is found to be superior to other methods like multi-layer perceptron (MLP), Random Forest, decision tree, nearest neighbor, and Naive Bayes. In another study, [6], nearest neighbor is found to be superior to other evaluated methods like logistic regression, Naive Bayes, KNN, SVM, Random Forest, and Stacking. This is interesting and not at all surprising, since many machine learning methods heavily depend on the initialization process and the difference of performance may be due to the random initialization process.

There are several datasets that are publically available for predicting cardiovascular disease. Some studies compare the effectiveness of the methods with several different public datasets. In [7],two datasets are used the Arrhythmia UCI and the Cardiovascular Kaggle datasets. Similarly, two datasets are used in [6], They are Cleveland and Cardiovascular datasets. In [8], three heart disease-related datasets are applied to the Decision Tree, SVM, KNN, Naive Bayes, Random Forest, Logistic Regression, and Majority Voting Classifier algorithms. Utilize manually created processes in all of those studies to improve performance further.

While the implementations of machine learning show some promising results, the performance may not yet give satisfactory results [3], [5], [6], [7], [9].One possible reason is the high volume of data, heterogeneity and complexity con-

*E-mail address: 14002338@nusamandiri.ac.id, hilm003@brin.go.id, agus@nusamandiri.ac.id*

tained in medical data. Many machine learning techniques have been used on a dataset of cardiovascular disease, but this is a difficult problem to solve. In traditional machine learning methods, i.e. non deep learning, handcrafted features are designed to deal with data complexity and nonlinearity. However, this requires a human labor that is expensive and depends on a master's knowledge, and it's not generalized properly.

Currently, there is an increasing trend in machine learning called deep learning. Deep learning is usually applied in supervised learning for various types of data from image [10], [11], [12], text [13], [14], [15], audio [16], [17], [18] or video [19], [20], [21]. Now, it is also used for unsupervised learning such as implementation of feature learning. In feature learning, the deep learning methods are designed such that they learn the underlying latent variables on the data. Feature learning is used to detect features or classifications of raw data to automatically find needed representation Machine learning such as classification often requires mathematical input and easily processed computing become the basic for feature learning. Feature learning eliminates manual engineering of features, allowing machines to learn specific tasks using those features, and study the features themselves to learn "how to learn". The predictive model architecture can be made simpler and prediction performance can be increased by using feature learning to learn efficient and succinct feature representations. [22].

The autoencoder is one of an artificial neural network's unsupervised learning algorithms. The autoencoder is trained to produce an output that closely resembles the original input. The three layers that make up the autoencoder are input, hidden, and output. The hidden layer (bottleneck) has smaller dimensions than the input layer, while the output layer is called the reconstruction layer. To train the system to reconstruct the input, the same number of neurons are present in both the input and output layers.[23]. The autoencoder is used to find new input representations without losing too much information and the input can be reconstructed [24]. Inputs in autoencoder can be reconstructed effectively with minimum reconstruction error [25]. Some researchers use autoencoder as a learning feature because it performs both linear and non-linear projections and outperforms PCA in the processing of complex, high-dimensional datasets [26], [27]. Akkalakshmi [28], using Autoencoder in predicting cancer disease to determine latent features. The autoencoder is tuned with various optimizers and batch sizes. This study concludes that the autoencoder used to determine latent features in the built neural network classifier shows a good increase in accuracy and produces low variance. Yousefi-Azar [29], Use Autoencoder to classify malware, detect network-based anomalies, and detect cyber threats. To investigate the latent representation of various feature sets, autoencoder is employed as a generative model. This study compares the classifier model using the original features with the classifier model using an autoencoder. Based on these results, the Gaussian Naive Bayes, SVM and XGBoost classifier models using an autoencoder are superior to the original feature.

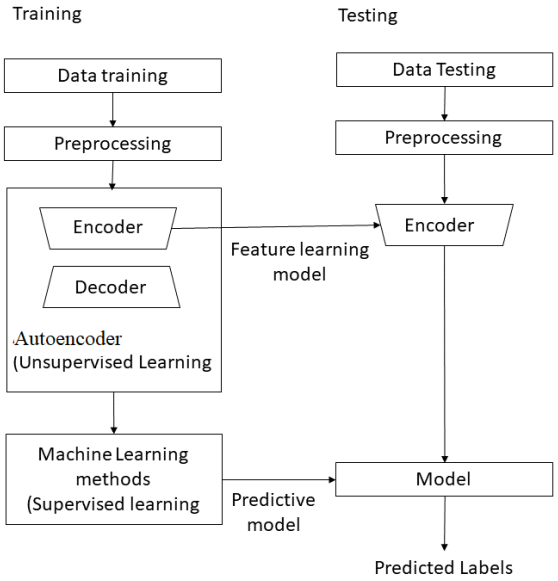


Figure 1. Implementations of autoencoder as feature learning

In this paper, we contribute to implementing feature learning in detecting Cardiovascular disease using Autoencoder. The complexity and underlying correlation of the heart disease data could be learned by an autoencoder. We use the output of the encoder of the autoencoder as inputs to several classifiers following unsupervised training of the autoencoder. They are Logistic Regression, Naive Bayes, Support Vector Machine, K-Nearest Neighbors, Decision Tree, XGBoost, Random Forest, and Neural Network. The outcome demonstrates an improvement in performance and shows promise.

## 2. RESEARCH METHOD

In the preparation of this research, several stages are needed to achieve the goals set previously. The training phase and the testing phase are the two stages of research in cardiovascular disease prediction. A learning feature called an autoencoder will be used to train the data during the training phase. An encoder and a decoder make up the autoencoder. The encoder develops the ability to decode input and compress it to an internal representation that is determined by the bottleneck layer. The decoder attempts to reconstruct the input by using the encoder's output (the bottleneck layer). After training, the autoencoder only retains the trained encoder, which is then used in supervised learning techniques to create a predictive model. This phase produces a model that will be used for testing or evaluation. While in the testing phase, the data will be evaluated based on the encoder from the feature learning model and the model from the predictive model so that in this phase it produces data with predicted labels. The flow of the research stages in predicting cardiovascular disease can be seen in Figure 1.

### A. Dataset

The dataset used in this study was downloaded from Kaggle using the reference [30]. There are 70,000 cardiovascular patient data records in the dataset, which has 12 features and 1 target feature and a file size of 2 point 94 MB. There are three different categories of input features: factual data, medical test results, and patient-provided data. The dataset's characteristics include 5 numerical attributes—age, height, weight, and systolic blood pressure—as well as 4 binary features—smoking, alcohol consumption, physical activity, and the presence or absence of cardiovascular disease—as well as 3 categorical features. (ap_lo), as well as diastolic blood pressure (ap_hi). The features of the dataset used in this study are presented in Table I while the samples of the first 5 of the dataset are presented in Table II

### B. Data Preprocessing

In order to ensure that the dataset used for training and testing is made up of high-quality data that is free of noise and bias, data preparation is done at this stage. Several things were done in data preprocessing, including removing data duplication, removing the 'id' feature, changing the feature age, eliminating outliers, adding BMI features, tension and removing the height, weight, ap_hi and ap_lo features. The details of preprocessing techniques are as follows:

1) Removing duplicate data: deleted instances related to data duplication as many as 24 instances.
2) Deleting Feature 'id': the feature is deleted because it has a very low correlation level.
3) Changed Feature 'age' from a matter of days to a matter of years. Additionally, it is divided into 4 classes. Class 0 (age <= 39), Class 1 (age 40–49), Class 2 (age 50–59), and Class 3 (age equal to or greater than 60).
4) Eliminating outliers in the height and weight features: outliers were removed from a number of features by first analyzing their scatter plots and boxplots Figure 2 illustrates some outliers for both features. Let's say the minimum age is 10798 days (29 years), the minimum height is 55 cm, and the minimum weight is 10 kg. In this case, there could be data entry errors. The highest weight is 200 kg, and the highest height is 250 cm, which may not be significant. Therefore, adjustments must be made to prevent this. BMI features are added before the outliers are eliminated. This new feature eliminates data that meets the criteria of being unnaturally very thin or very fat by dividing body weight by height square. For example, there is an instance with a height of 80 cm and a weight of 178 kg. The instance may be an input error. So to overcome this, data with BMI < 5 and > 100 features are omitted. The results are then divided into 4 groups: class 0 (BMI < 18.5), class 1 (BMI 18.5–24.9), class 2 (BMI 25.0–29.9) and class 3 (BMI more than equal to 30.0). Additionally, the features of height and weight have been eliminated since the BMI feature already includes these two characteristics.
5) Added a "tension" feature to help readers understand the Systolic Blood Pressure (ap_hi) and Diastolic Blood Pressure (ap_lo) features and to help classify the ap_hi and ap_lo. features that are used to measure a patient's blood pressure or detect hypertension.. In both features, instances exceeding 500 mmHg are removed, and the ap_hi feature must be greater than the ap_lo feature. Furthermore, the 'tension' feature is categorized into 6 classes, namely class 0 (low blood pressure – with ap_hi less than 80 or ap_lo less than 60), class 1 (normal – with ap_hi 80 to 120 and ap_lo 60 to 80), class 2 (prehypertension - with ap_hi 120 to 139 or ap_lo 80 to 89), class 3 (Hypertension Stage 1 - with ap_hi 140 to 159 or ap_lo 90 to 99), class 4 (Hypertension Stage 2 - with ap_hi 160 to 179 or ap_lo 100 to 109), and class 5 (High Blood Pressure Crisis - with ap_hi > 180 and ap_lo > 110). Furthermore, the features of ap_hi and ap_lo were removed, because these two features were already represented in the tension feature.

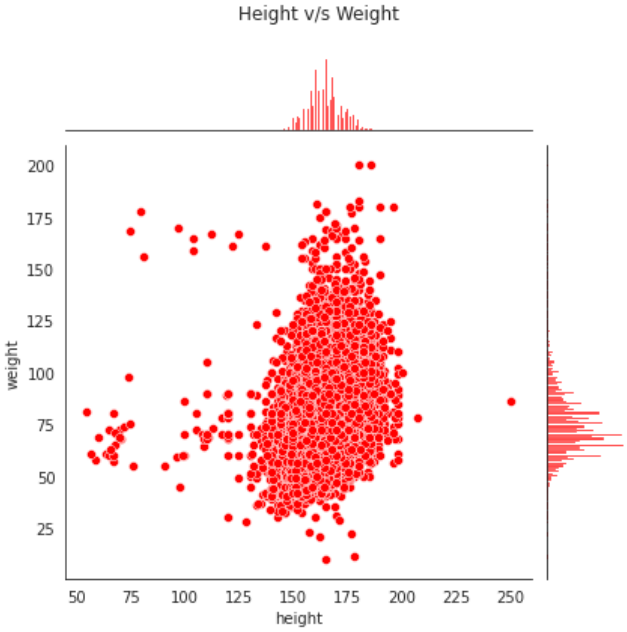


Figure 2. Distribution of Height and Weight

Following that, it is applied by one hot encoding to convert categorical data into binary data. Because category data don't actually have an order, the intention is to suggest greater accuracy [31] For coding methods employing the One Hot Encoding methodology, from the overall features predictor. In the binary converter method used by this methodology, only one bit of the condition variable is set to "1" or "hot" for each specific situation, while all other

TABLE I. Description features of dataset

| Feature | Description |
|---|---|
| id | patient id number |
| age | age (days) |
| height | height (cm) |
| weight | weight (kg) |
| gender | 1 or 2 |
| systolic blood pressure (aphi) | health check results (int) |
| diastolic blood pressure (aplo) | health check results (int) |
| cholesterol | 1: normal, 2: above normal, 3: well above normal |
| Glucose (gluc) | 1: normal, 2: above normal, 3: well above normal |
| Smoking (smoke) | 0: no smoke, 1: smoke |
| Alcohol intake (alco) | 0: no alco, 1: alco |
| Physical activity (active) | 0: no active, 1: active |
| Presence or absence of cardiovascular disease (cardio) | 0: no cardio, 1: cardio |

TABLE II. Samples of the first 5 of the dataset

| id | age | gender | height | weight | ap_hi | ap_lo | cholesterol | gluc | smoke | alco | active | cardio |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 18293 | 2 | 168 | 62.0 | 110 | 80 | 1 | 1 | 0 | 0 | 1 | 0 |
| 1 | 20228 | 1 | 156 | 85.0 | 140 | 90 | 3 | 1 | 0 | 0 | 1 | 1 |
| 2 | 18857 | 1 | 165 | 64.0 | 130 | 70 | 3 | 1 | 0 | 0 | 0 | 1 |
| 3 | 17623 | 2 | 169 | 82.0 | 150 | 100 | 1 | 1 | 0 | 0 | 1 | 1 |
| 4 | 17474 | 1 | 156 | 56.0 | 100 | 60 | 1 | 1 | 0 | 0 | 0 | 0 |

conditional bits are set to 0. The outcomes of the One Hot Encoding preprocessing are shown in Table III Following one hot encoding operation, the dataset is split into training and testing data. The dataset for this study is divided into training and testing data in an 8:2 ratio, with training data making up 80% and testing data 20% of the total (68.673). 13735 records make up the testing data compared to 54.938 records in the training set.

### C. Architecture of the Autoencoder

The Autoencoder is currently being used to learn features. By employing backpropagation and setting the target value equal to the input, the Autoencoder process is carried out using the encoder-decoder paradigm. The input is first changed into an encoder, which is a hidden representation with fewer dimensions than the input vector. In order to obtain the output of the reconstructed or generated network given input with high probability, the decoder extends or remaps the features extracted from the given input, which is referred to as the hidden representation. Because of a reconstruction error, the output autoencoder won't reconstruct the input accurately. The error function, which is typically a mean-squared error or cross-entropy, is used to punish the network when its output deviates from its input. The answer will depend on the size of the hidden representation or the extracted features. Reconstruction error increases with decreasing hidden representation dimension. Feature dimensions and information loss must now be balanced against one another. The aim is to keep the majority of the data's information while minimizing the feature's

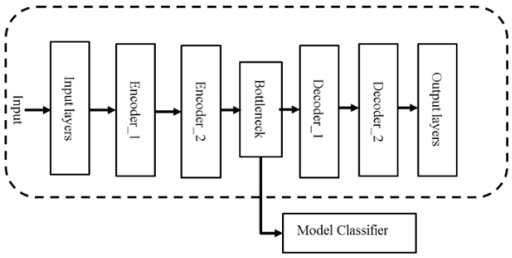


Figure 3. Architecture of the autoencoder

dimensions. After a layer has been trained, its output is passed on to the following layer to create a highly non-linear dependency model on the input. The dimensions of the input data are to be reduced by this process. The features that are extracted for classification are based on the layer encoded in the autoencoder's bottleneck (center). You can see the autoencoder model that was used in Figure 3.

The five models that make up the autoencoder model used in this study are as follows:

1) The Autoencoder model uses 3 layers: an input layer, a bottleneck, and an output layer make up this system.
2) The Autoencoder model uses 4 layers: it consists of an input layer, an encoder_1, a bottleneck, and an output layer.

TABLE III. Samples of the first 6 of one hot encoding's results

| age_0 | age_1 | age_2 | age_3 | gender_1 | gender_2 | ..... | bmi_1 | bmi_2 | bmi_3 | cardio |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 0 | 1 | ..... | 1 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 1 | 0 | ..... | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 1 | 0 | ..... | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 1 | ..... | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | ..... | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | ..... | 0 | 1 | 0 | 0 |

3) The Autoencoder model uses 5 layers: it consists of an input layer, an encoder_1, a bottleneck, a decoder_1, and an output layer.
4) The Autoencoder model uses 6 layers: it consists of an input layer, an encoder_1, an encoder_2, a bottleneck, a decoder_1, and an output layer.
5) The Autoencoder model uses 7 layers: It is made up of the following components: an input layer, an encoder 1, an encoder 2, a bottleneck, a decoder 1, a decoder 2, and an output layer.

28 input neurons are used in the autoencoder process, and encoder_1, encoder_2, decoder_1, and decoder_2 are converted into 28 neurons. At the bottleneck, X is repeatedly transformed into 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 14, 10, and 6 neurons to obtain the best accuracy value. According to how many neurons are present, the output from the bottleneck is transformed into an extracted feature. information that has been reduced to its smallest feature dimensions. The Model Classifier can be trained and tested using the output. An autoencoder was used to extract the dimensions, after which they were converted into features for 27, 26, 25, 24, 23, 22, 21, 20, 19, 18, 14, 10, and 6 neurons. The Autoencoder (AE) model is configured with the activation function at the bottleneck, encoder_1, encoder_2, decoder_1, and encoder_2 using Relu, the kernel initializer used is Random Normal with a standard deviation of 0.01, the initializer bias used is Zeros, and linear activation is used on the output layers. The experiment was run using the parameters epoch=500, batch size=1000, optimizer=Adam, and loss function=MSE.

### *D. Classifiers*

Now that the encoder process has been completed, the dataset is trained and tested using the Logistic Regression, Naive Bayes, Support Vector Machine, K-Nearest Neighbors, Decision Tree, XGBoost, Random Forest, and Neural Network algorithms using the trained encoder. the module used in this process uses Scikit-learn, the parameters used in Logistics regression and Naive Bayes are default from Scikit-learn, the parameters used in SVM are the RBF kernel, the parameters used in KNN number of neighbors = 50, the power parameter (p)= 1, weights='uniform', the parameters used in the Decision Tree are criterion = 'gini', random state=42, maximum depth of the tree= 10, the parameters used in XGB are verbosity= 0, seed= 0 , number of estimators= 150, gamma= 0.24, maximum depth of the tree= 4, learning rate=0.13, reg lambda= 50.0, scale pos weight= 1, the parameter used in Random Forest is random state= 42, number of estimators= 100, maximum depth of the tree= 10, criterion = 'entropy', parameters used in Neuron Network are activation='relu', kernel initializer = Random Normal (stddev= 0.01, seed= None), bias initializer= Zeros, kernel regularizer=l2(0.0001).

### *E. Evaluation*

At this point, each Classifier model is evaluated. the model will be assessed through data testing. Scikit-learn libraries are used during the evaluation process. Performance metrics used to assess the model's efficacy include accuracy, precision, recall, and f1-score. The proposed model is picked by comparing and contrasting the performance metric results.

At this stage, all neuron variants from the Autoencoder layer are compared to get the best accuracy from each classifier model, then the best accuracy is compared to the baseline, so that it can be seen whether the classifier model using the autoencoder as Feature Learning can increase accuracy from the baseline. The averages of the classifier models, neurons, and autoencoder layers were also compared. These comparisons were used to determine whether the autoencoder layer was shallower or deeper and to analyze the pattern of the applied autoencoder architecture. actually, or better. the classifier model's accuracy decreases.

## 3. RESULTS AND DISCUSSIONS

Before performing the Autoencoder process, all predictor features are converted into one hot encoding technique. To determine the level of classification accuracy, the output of feature learning is assessed using a classifier model that includes Logistic Regression, Naive Bayes, Support Vector Machine, K-Nearest Neighbors, Decision Tree, XGBoost, Random Forest, and Neural Network. These results will be compared with the classifier model without using Feature Learning (baseline). This is done to evaluate whether the Autoencoder as a Learning Feature can increase accuracy from the baseline. Each layer of the Autoencoder takes the highest accuracy value for each classifier model, then the accuracy value is compared with the classifier model without using the Autoencoder (baseline). The following is the classification result of the classifier model that is applied using the Autoencoder as a Learning Feature compared

to the classifier model without using the Autoencoder (Baseline) as can be seen in Table IV and Figure 5.

Based on the comparison of the accuracy of Table IV and Figure 5, it can be seen that the classifier model using an autoencoder can improve the accuracy of the classifier model without an autoencoder (baseline). The highest increase in accuracy is found in the Naive Bayes classifier model of 0.75 % even though the Naive Bayes accuracy is smaller than other classifier models. The lowest increase in accuracy is found in the Neural Network classifier model of 0.001 %. Based on these results, it shows that the Autoencoder is able to improve accuracy even though the results are not high, so further research is needed. Confusion matrix can be seen in Figure 4. Based on the confusion matrix's findings, it can be shown that using an autoencoder to predict individuals who actually have cardiovascular disease is more accurate than baseline, such as logistic regression, naive Bayes, KNN, decision trees, random forests, and neural networks. While using an autoencoder to predict patients who do not have cardiovascular disease is more accurate than the baseline in the SVM and XGB method.

In addition, to analyze the pattern of the autoencoder model, the average comparison between neurons, between classifier models, and between autoencoder layers is carried out. This comparison is used to analyze whether the shallower or deeper layers of the autoencoder accuracy produced, the better or worse the accuracy of the classifier model.

### A. *Analyzing the average accuracy of bottleneck neurons*

Based on the comparison of the average accuracy between neurons of bottleneck (Table V),it can be seen that the highest average accuracy in the Autoencoder with 3 layers is in the bottleneck with 23 neurons of 72.20, in the Autoencoder with 4 layers the highest average accuracy found in the bottleneck with 26 and 25 neurons of 72.38, in the Autoencoder with 5 layers the highest average accuracy found in the bottleneck with 22 neurons of 72.27, in Autoencoder with 6 layers the highest average accuracy found in a bottleneck with 24 neurons of 72.40, in Autoencoder with 7 layers the highest average accuracy is found in a bottleneck with 27 neurons of 71.98, so that from the overall average comparison between neurons, the highest average accuracy is found in the Autoencoder with 6 layers with 24 neurons. The graph of the average comparison between neurons can be seen in Figure 4.

### B. *Analyzing the average accuracy of various classifier models*

On the average between classifier models it can be seen that (Table VI and Figure 7), The highest average accuracy for the logistic regression algorithm is found in the Autoencoder 4 layers of 72.40; for the Naive Bayes algorithm, it is found in the Autoencoder 4 layers of 70.82; for the SVM algorithm, it is found in the Autoencoder 3 layers of 72.59; for the KNN algorithm, it is found in the Autoencoder 4 layers of 72.04; for the Decision Tree

| Baseline | | | | | Autoencoder | | | | |
|---|---|---|---|---|---|---|---|---|---|
| LR | | | Predicted | | LR | | | Predicted | |
| | | | 0 | 1 | | | | 0 | 1 |
| | Actual | 0 | 5541 | 1440 | | Actual | 0 | 5536 | 1445 |
| | | 1 | 2315 | 4439 | | | 1 | 2298 | 4456 |
| NB | | | Predicted | | NB | | | Predicted | |
| | | | 0 | 1 | | | | 0 | 1 |
| | Actual | 0 | 5351 | 1630 | | Actual | 0 | 5428 | 1553 |
| | | 1 | 2342 | 4412 | | | 1 | 2316 | 4438 |
| SVM | | | Predicted | | SVM | | | Predicted | |
| | | | 0 | 1 | | | | 0 | 1 |
| | Actual | 0 | 5377 | 1604 | | Actual | 0 | 5387 | 1594 |
| | | 1 | 2147 | 4607 | | | 1 | 2150 | 4604 |
| KNN | | | Predicted | | KNN | | | Predicted | |
| | | | 0 | 1 | | | | 0 | 1 |
| | Actual | 0 | 5541 | 1440 | | Actual | 0 | 5367 | 1614 |
| | | 1 | 2362 | 4392 | | | 1 | 2178 | 4576 |
| DT | | | Predicted | | DT | | | Predicted | |
| | | | 0 | 1 | | | | 0 | 1 |
| | Actual | 0 | 5285 | 1696 | | Actual | 0 | 5249 | 1732 |
| | | 1 | 2090 | 4664 | | | 1 | 2049 | 4705 |
| XGB | | | Predicted | | XGB | | | Predicted | |
| | | | 0 | 1 | | | | 0 | 1 |
| | Actual | 0 | 5341 | 1640 | | Actual | 0 | 5422 | 1559 |
| | | 1 | 2093 | 4661 | | | 1 | 2163 | 4591 |
| RF | | | Predicted | | RF | | | Predicted | |
| | | | 0 | 1 | | | | 0 | 1 |
| | Actual | 0 | 5336 | 1645 | | Actual | 0 | 5296 | 1685 |
| | | 1 | 2121 | 4633 | | | 1 | 2069 | 4685 |
| NN | | | Predicted | | NN | | | Predicted | |
| | | | 0 | 1 | | | | 0 | 1 |
| | Actual | 0 | 5343 | 1638 | | Actual | 0 | 5258 | 1723 |
| | | 1 | 2105 | 4649 | | | 1 | 2019 | 4735 |

Figure 4. Confusion Matrix of model classifier using AE with baseline


Figure 5. Comparison of model classifier using AE with baseline

TABLE IV. Comparison of model classifier using ae with baseline

| | LR | NB | SVM | KNN | DT | XGB | RF | NN |
|---|---|---|---|---|---|---|---|---|
| Baseline | 72,66 | 71,08 | 72,69 | 72,32 | 72,44 | 72,82 | 72,58 | 72,75 |
| Autoencoder | 72,75 | 71,83 | 72,74 | 72,39 | 72,47 | 72,90 | 72,67 | 72,76 |
| Increase of accuracy | 0,09 | 0,75 | 0,05 | 0,07 | 0,03 | 0,08 | 0,09 | 0,01 |

TABLE V. Analyzing the average accuracy of bottleneck neurons

| Num of neuron in bottleneck | Autoencoder 3 layers | Autoencoder 4 layers | Autoencoder 5 layers | Autoencoder 6 layers | Autoencoder 7 layers |
|---|---|---|---|---|---|
| 27 | 72,16 | 72,26 | 72,16 | 72,26 | **71,98** |
| 26 | 72,12 | **72,38** | 72,06 | 72,02 | 70,95 |
| 25 | 72,13 | **72,38** | 72,01 | 72,07 | 70,82 |
| 24 | 72,17 | 72,29 | 72,20 | **72,40** | 70,99 |
| 23 | **72,20** | 71,87 | 72,22 | 71,96 | 70,48 |
| 22 | 72,13 | 72,20 | **72,27** | 72,36 | 69,50 |
| 21 | 72,17 | 72,36 | 71,82 | 71,14 | 71,96 |
| 20 | 72,14 | 72,26 | 71,19 | 70,35 | 71,05 |
| 19 | 72,15 | 72,33 | 70,38 | 72,32 | 71,88 |
| 18 | 71,93 | 72,27 | 71,88 | 71,10 | 71,36 |
| 14 | 71,96 | 72,20 | 71,56 | 71,42 | 71,12 |
| 10 | 71,87 | 72,01 | 69,79 | 70,24 | 70,60 |
| 6 | 71,74 | 71,78 | 70,73 | 68,87 | 68,99 |

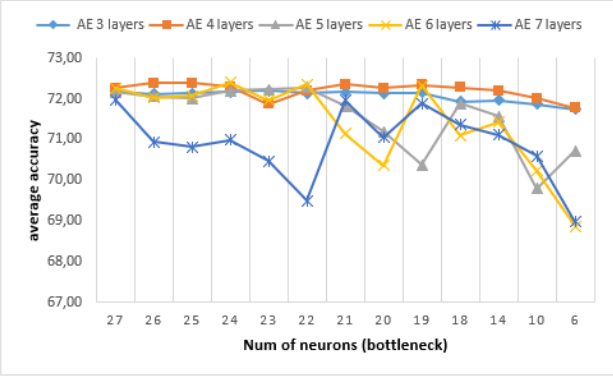


Figure 6. Analyzing the average accuracy of bottleneck neurons

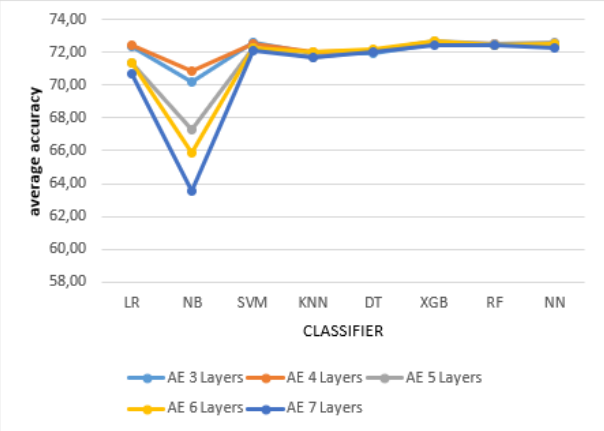


Figure 7. Analyzing the average accuracy of various classifier models

algorithm, it found in the Autoencoder 6 layers at 72.18; for the XGBoost algorithm, it is found in Autoencoder 5 layers at 72.69; for the Random Forest algorithm, it is found in Autoencoder 5 layers at 72.52; and for the Neural Network algorithm, its found in Autoencoder 5 layers at 72.62.

Accordingly, when comparing the average accuracy of different classifier models, the SVM classifier model performs best on the autoencoder with 3 layers, followed by the LR, NB, and KNN classifier models on the autoencoder with 4 layers, and the XGB, RF, and NN classifier models on the autoencoder with 5 layers. the DT classifier model achieves the highest accuracy on the autoencoder with 6 layers.

### C. *Analyzing the average accuracy of various autoencoder layers*

On the average between layers of the autoecoder (Figure 8), it can be seen that, for autoencoder 3 layers, the average accuracy is 72.07, for autoencoder 4 layers, the average accuracy is 72.20, for autoencoder 5 layers, the average accuracy is 71.56, for autoencoder 6 layers, the average accuracy is 71.42, while in autoencoder 7 layers, the average accuracy is 70.90. So from the comparison of the average accuracy between the autoencoder layers, the highest average accuracy is found in the Autoencoder 4 layers and the lowest accuracy is in the Autoencoder 7

TABLE VI. Analyzing the average accuracy of various classifier models

| Model Autoencoder | LR | NB | SVM | KNN | DT | XGB | RF | NN |
|---|---|---|---|---|---|---|---|---|
| Autoencoder 3 Layers | 72,38 | 70,16 | **72,59** | 71,94 | 71,95 | 72,53 | 72,45 | 72,55 |
| Autoencoder 4 Layers | **72,40** | **70,82** | 72,55 | **72,04** | 72,10 | 72,64 | 72,50 | 72,54 |
| Autoencoder 5 Layers | 71,35 | 67,32 | 72,24 | 71,66 | 72,07 | **72,69** | **72,52** | **72,62** |
| Autoencoder 6 Layers | 71,38 | 65,90 | 72,26 | 72,03 | **72,18** | 72,66 | 72,47 | 72,49 |
| Autoencoder 7 Layers | 70,72 | 63,55 | 72,09 | 71,66 | 72,01 | 72,45 | 72,41 | 72,30 |

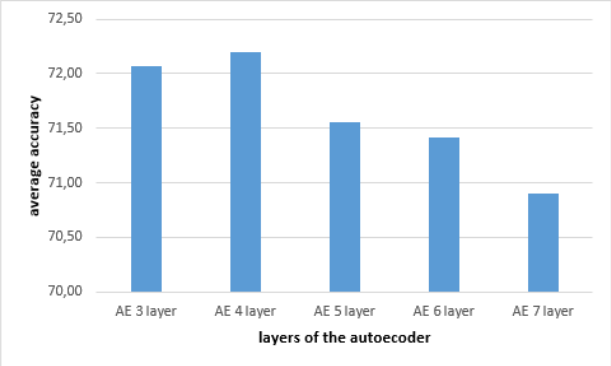


Figure 8. Analysis of the average accuracy of the various autoencoder layers

layers.

Based on the description above, it can be seen that the shallower or deeper layers in the autoencoder do not make the accuracy better. This is based on the results of the average accuracy generated as follows: In the average comparison between neurons in the bottleneck, the highest average accuracy is found in the Autoencoder 6 layers with 24 neurons, In a comparison of the average accuracy between the classifier models, the autoencoder 4 model layers and 5 layers autoencoder, there are 3 classifier models that get the highest accuracy, in the 3 layers autoencoder and 6 layers autoencoder models there is 1 classifier model that gets the highest accuracy, while the 7 layers autoencoder does not have a classifier model that gets the highest accuracy. While the comparison of the average accuracy between layers of autoencoder, the highest average accuracy is found in the 4 layers autoencoder model and the lowest average accuracy is in the 7 layers autoencoder model.

## 4. CONCLUSIONS

Autoencoder architecture can be implemented in predicting cardiovascular disease. This architecture is able to increase the accuracy of the classifier model used in the study. Accuracy in the Logistic Regression algorithm is 72.75%, an increase of 0.09% from the baseline using Autoencoder 4 Layers with 24 neurons, accuracy in the Naive Bayes algorithm is 71.83%, an increase of 0.75% from the baseline using Autoencoder 4 Layers with 26 neurons, accuracy in the Support Vector Machine algorithm is 72.74% increased 0.05% from baseline using Autoencoder 3 Layers with 20 neurons, accuracy in the K-Nearest Neighbors algorithm was 72.39% increased 0.07% from baseline using Autoencoder 4 Layers with 18 neurons, accuracy in Decision Tree algorithm was 72.47% increased 0.03% from baseline using Autoencoder 6 Layers with 18 neurons, accuracy in XGBoost algorithm is 72.9%, an increase of 0.08% from baseline using Autoencoder 5 Layers with 20 neurons and Autoencoder using 6 Layers with 22 neurons, accuracy in Random Forest algorithm is 72.67% an increase of 0.09% from baseline using Autoencoder 4 Layers with 25 neurons and autoencoder 5 Layers with 23 neurons while the accuracy of the Neural Network algorithm is 72.76%, an increase of 0.01% from the baseline using Autoencoder 5 Layers with 26 neurons.

This study found that, if you compare the average accuracy values, both the average accuracy between neurons at the bottleneck, the average accuracy between the classifier models and the average accuracy between the Autoencoder Layers, it is known that the shallower or deeper layers on the autoencoder do not make accuracy better. This is based on the results of the average accuracy produced as follows: In the average comparison between neurons in the bottleneck, the highest average accuracy is found in the Autoencoder 6 layers with 24 neurons of 72.40, in the comparison of the average accuracy between the classifier models, the 4 layers autoencoder and 5 layers autoencoder models, there are 3 classifier models that get the highest accuracy, in the 3 layers autoencoder and 6 layers autoencoder models there is 1 classifier model that gets the highest accuracy, while the 7 layers autoencoder does not. There is a classifier model that gets the highest accuracy, while in the comparison of the average accuracy between layers of the autoencoder, the highest average accuracy is in the 4-layer autoencoder model of 72.20 and the lowest average accuracy is found in the 7-layer autoencoder model of 70.90. Based on these results, it can be said that the autoencoder can increase the performance model's accuracy, even though the accuracy results obtained are not very high. It can also be said that the layer depth does not always enhance the performance model's accuracy. This is because the primary goal of the paper is to investigate the effects of feature learning using an autoencoder, as well as the effects of changing the autoencoder's layer structure. These outcomes create possibilities for future development. They can improve the accuracy of their cardiovascular disease prediction by using other autoencoder models like the convolutional autoencoder, variational autoencoder, and other autoencoder variants.

## ReferenceIJCDS


[1] M. F. Zhuhri, "Menko pmk: Prevalensi penyakit jantung terus meningkat," *https://mediaindonesia.com/humaniora/411528/menko-pmk-prevalensi-penyakit-jantung-terus-meningkat*, 2021.

[2] K. V. S. A.Geetha Devi, Surya Prasada Rao Borra, "A method of cardiovascular disease predictionusing machine learning," *International Journal of Engineering Research Technology (IJERT)*, vol. 9, no. 5, 2021.

[3] D. P. F. D. G. M. Dr. Rakhi Waigi, Dr. Sonali Choudhary, "Predicting the risk of heartdisease using advanced machinelearning approach," *European Journal of Molecular Clinical Medicine*, vol. 7, no. 07, 2020.

[4] A. R. S. R. Halfis, "Prediction and analysis of cardiovascular disease with neural network algorithm," *Jurnal TECHNO Nusa Mandiri*, vol. 16, no. 2, 2019.

[5] A. Duarte and O. Belo, "Cardiac well-being indexes: a decision support tool to monitor cardiovascular health," *Journal of Integrative Bioinformatics*, vol. 18, no. 2, pp. 127–138, mar 2021.

[6] D. Kumar, "Cardiovascular disease prediction using machine learning," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 46–54, sep 2020.

[7] R. Hagan, C. J. Gillan, and F. Mallett, "Comparison of machine learning methods for the classification of cardiovascular disease," *Informatics in Medicine Unlocked*, vol. 24, p. 100606, 2021.

[8] M. I. H. Khan and M. R. H. Mondal, "Data-driven diagnosis of heart disease," *International Journal of Computer Applications*, vol. 176, no. 41, pp. 46–54, jul 2020.

[9] M. Padmanabhan, P. Yuan, G. Chada, and H. V. Nguyen, "Physician-friendly machine learning: A case study with cardiovascular disease risk prediction," *Journal of Clinical Medicine*, vol. 8, no. 7, p. 1050, jul 2019.

[10] P. Patel and A. Thakkar, "The upsurge of deep learning for computer vision applications," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, p. 538, feb 2020.

[11] S. Bhattacharya, P. K. R. Maddikunta, Q.-V. Pham, T. R. Gadekallu, S. R. K. S, C. L. Chowdhary, M. Alazab, and M. J. Piran, "Deep learning and medical image processing for coronavirus (COVID-19) pandemic: A survey," *Sustainable Cities and Society*, vol. 65, p. 102589, feb 2021.

[12] X. Jin, J. Che, and Y. Chen, "Weed identification using deep learning and image processing in vegetable plantation," *IEEE Access*, vol. 9, pp. 10 940–10 950, 2021.

[13] A. Wahdan, S. A. Hantoobi, S. A. Salloum, and K. Shaalan, "A systematic review of text classification research based on deep learning models in arabic language," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 6, p. 6629, dec 2020.

[14] A. Onan, "Sentiment analysis on massive open online course evaluations: A text mining and deep learning approach," *Computer Applications in Engineering Education*, vol. 29, no. 3, pp. 572–589, may 2020.

[15] S. C, S. Adak, and S. C. Tigga, "Opinionated text classification for hindi tweets using deep learning," in *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*. IEEE, apr 2021.

[16] H. F. Pardede, A. R. Yuliani, and R. Sustika, "Convolutional neural network and feature transformation for distant speech recognition," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 8, no. 6, p. 5381, dec 2018.

[17] R. A. Ramadan, "Detecting adversarial attacks on audio-visual speech recognition using deep learning method," *International Journal of Speech Technology*, vol. 25, no. 3, pp. 625–631, jun 2021.

[18] L. Schoneveld, A. Othmani, and H. Abdelkawy, "Leveraging recent advances in deep learning for audio-visual emotion recognition," *Pattern Recognition Letters*, vol. 146, pp. 1–7, jun 2021.

[19] A. Ignatov, A. Romero, H. Kim, R. Timofte, C. M. Ho, Z. Meng, K. M. Lee, Y. Chen, Y. Wang, Z. Long, C. Wang, Y. Chen, B. Xu, S. Gu, L. Duan, W. Li, W. Bofei, Z. Diankai, Z. Chengjian, L. Shaoli, G. Si, Z. Xiaofeng, L. Kaidi, X. Tianyu, Z. Hui, X. Gao, X. Wang, J. Guo, X. Zhou, H. Jia, and Y. Yan, "Real-time video super-resolution on smartphones with deep learning, mobile ai 2021 challenge: Report," 2021.

[20] L. Jiao, R. Zhang, F. Liu, S. Yang, B. Hou, L. Li, and X. Tang, "New generation deep learning for video object detection: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 8, pp. 3195–3215, aug 2022.

[21] B. Liu, Y. Chen, S. Liu, and H.-S. Kim, "Deep learning in latent space for video prediction and compression," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2021.

[22] M. Ding, K. Yang, D.-Y. Yeung, and T.-C. Pong, "Effective feature learning with unsupervised learning for improving the predictive models in massive open online courses," in *Proceedings of the 9th International Conference on Learning Analytics &amp Knowledge*. ACM, mar 2019.

[23] S. A. Ebiaredoh-Mienye, E. Esenogho, and T. G. Swart, "Artificial neural network technique for improving prediction of credit card default: A stacked sparse autoencoder approach," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 5, p. 4392, oct 2021.

[24] H. Nugroho, M. Susanty, A. Irawan, M. Koyimatu, and A. Yunita, "Fully convolutional variational autoencoder for feature extraction of fire detection system," *Jurnal Ilmu Komputer dan Informasi*, vol. 13, no. 1, p. 9, mar 2020.

[25] M. Liang, R. W. Liu, S. Li, Z. Xiao, X. Liu, and F. Lu, "An unsupervised learning method with convolutional auto-encoder for vessel trajectory similarity computation," *Ocean Engineering*, vol. 225, p. 108803, apr 2021.

[26] J. M. A. K. V. K. D. R. B, "Optimized medical disease analysis usingautoencoder and multilayer perceptron," *IJCSN - International Journal of Computer Science and Network*, vol. 8, no. 3, 2019.

[27] S. C. Chae and S.-Y. Choi, "Analysis of the term structure of major currencies using principal component analysis and autoencoders," *Axioms*, vol. 11, no. 3, p. 135, mar 2022.

[28] M. Akkalakshmi, R. Y Md, V. Revathi, and A. Pal, "Auto encoder based feature learning and up-sampling to enhance cancer predic-

tion," *International Journal of Future Generation Communication and Networking*, pp. 1453–1459, 04 2020.

[29] M. Yousefi-Azar, V. Varadharajan, L. Hamey, and U. Tupakula, "Autoencoder-based feature learning for cyber security applications," in *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, may 2017.

[30] S. ulianova, "Cardiovascular disease dataset," *https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset*, 2019.

[31] Y. N. Kunang, S. Nurmaini, D. Stiawan, A. Zarkasi, Firdaus, and Jasmir, "Automatic features extraction using autoencoder in intrusion detection system," in *2018 International Conference on Electrical Engineering and Computer Science (ICECOS)*. IEEE, oct 2018.

**Angelina Puput Giovani** Angelina Study in Nusa Mandiri University, Jakarta-Indonesia. Her research interest lies in data mining and machine learning. She can be contacted at email: 14002338@nusamandiri.ac.id.

**Hilman Ferdinandus Pardede** Hilman is a research professor at the national agency for research and innovation in Indonesia. The Nusa Mandiri University also employs him as a professor. He graduated from the University of Indonesia with a bachelor's degree. He received his master's degree from Western Australia University. He received his doctorate in computer science from Tokyo Institute of Technology. Speech and signal processing, pattern recognition, and machine learning are some of his research specialties. Email him at hilm003@brin.go.id

**Agus Subekti** Lecturer at Nusa Mandiri University, Agus. At the Indonesian National Research and Innovation Agency's Research Center for Telecommunications, he also works as a senior researcher. From 2002 to 2004, he worked as a research associate at Tokai University's Nakajima Laboratory in Japan. He earned his electrical engineering and informatics bachelor's, master's, and doctoral degrees from Institut Teknologi Bandung (ITB). Machine learning, signal processing, and communication systems are some of his areas of interest in research. Email: agus@nusamandiri.ac.id.