International Journal of Contemporary Research in Computer Science and Technology (IJCRCST) e-ISSN: 2395-5325
Volume 8, Issue 1 (April '2025)

ENHANCING PREDICTIVE PERFORMANCE OF HEART DISEASE DETECTION USING XGBOOST AND AUTOENCODERS ON THE CARDIOVASCULAR DISEASE DATASET

M. Ranjani,
Research Scholar,
Department of Computer Science,
Periyar University,
Salem, TamilNadu, India.

Dr. P.R.Tamilselvi,
Assistant Professor,
Department of Computer Science,
Government Arts and Science College
(Affilated to Periyar University),
Komarapalayam,TamilNadu, India.

Abstract: Cardiovascular diseases (CVDs) remain the leading cause of mortality worldwide, emphasizing the urgent need for accurate and early detection methods. This research aims to enhance the predictive performance of heart disease detection by leveraging two powerful machine learning techniques: Extreme Gradient Boosting (XGBoost) and Autoencoders. Using the publicly available Cardiovascular Disease Dataset, we develop and evaluate models based on both techniques, with a focus on key classification metrics—Accuracy, Precision, Recall (Sensitivity), Specificity, and F1 Score. The XGBoost model is trained directly on the dataset, while the Autoencoder is employed for both anomaly detection and feature extraction. The study compares the performance of these models and explores the potential of a hybrid approach combining Autoencoder-based features with XGBoost classification. Results indicate that both models show promise, with XGBoost achieving high classification accuracy and Autoencoders contributing to enhanced feature representation. This work contributes to the growing field of AI-assisted medical diagnostics and offers insights into model selection for heart disease prediction tasks.

Keywords: *Heart Disease Detection, Cardiovascular Disease Dataset, XGBoost, Autoencoders, Machine Learning, Medical Diagnosis, Feature Extraction, Accuracy, Precision, Recall, Specificity, F1 Score*

I. INTRODUCTION

Cardiovascular diseases (CVDs) are the leading cause of death globally, accounting for an estimated 17.9 million lives lost each year according to the World Health Organization (WHO). These diseases encompass a range of heart and blood vessel disorders, with coronary artery disease and heart attacks being among the most prevalent. The early and accurate detection of heart disease is critical for effective treatment and prevention, potentially reducing mortality rates and improving patient outcomes. Traditional diagnostic methods, while effective, often involve expensive procedures or may not detect subtle risk patterns in time. In recent years, machine learning (ML) techniques have gained significant attention in the medical community for their ability to analyze complex datasets and uncover hidden patterns associated with diseases. Among these techniques, ensemble learning models like Extreme Gradient Boosting (XGBoost) and deep learning approaches such as Autoencoders have shown promise in various healthcare applications, including disease classification and anomaly detection.

XGBoost is a scalable and efficient implementation of gradient boosting that is particularly well-suited for structured data and tabular datasets. Its ability to handle missing values, perform automatic feature selection, and manage overfitting through regularization makes it an attractive choice for clinical data analysis. On the other hand, Autoencoders, a type of unsupervised neural network, are powerful for dimensionality reduction and anomaly detection. By learning compact, meaningful representations of input data, Autoencoders can enhance the performance of downstream classifiers or independently identify abnormal patterns indicative of disease. This study focuses on enhancing the predictive performance of heart disease detection using both XGBoost and Autoencoders on the Cardiovascular Disease Dataset. The objective is to assess and compare the models based on key classification metrics: Accuracy, Precision, Recall (Sensitivity), Specificity, and F1 Score. We also investigate the effectiveness of using Autoencoders for feature extraction to improve the performance of XGBoost. The motivation behind this research is to develop reliable and interpretable models that can aid healthcare professionals in making timely and accurate diagnostic decisions. By evaluating and comparing these techniques, we aim to identify robust approaches that can be integrated into clinical decision support systems for early heart disease detection.

II. RELATED WORKS

Heart disease prediction has been a significant focus in medical data science due to the high global mortality associated with cardiovascular conditions. With the advancement of machine learning (ML), various algorithms have been employed to improve diagnostic accuracy and support early detection.

Numerous studies have explored classical ML algorithms such as Decision Trees, Support Vector Machines (SVM), Naïve Bayes, and k-Nearest Neighbors (k-NN) for heart disease classification. For example, Singh et al. (2016) used Decision Trees and SVMs on the UCI Heart Disease dataset and found SVMs to outperform other models in terms of classification accuracy. Similarly, Palaniappan and Awang

(2008) demonstrated the effectiveness of Naïve Bayes and Neural Networks, though they acknowledged limitations in handling complex non-linear relationships in clinical data.

XGBoost has emerged as a highly effective ML technique in structured healthcare data tasks. Chen and Guestrin (2016) introduced XGBoost as a scalable and regularized boosting algorithm, which has since been adopted in medical diagnostics due to its robustness and performance. In a study by Tiwari et al. (2021), XGBoost outperformed Random Forest and Logistic Regression in predicting heart disease, achieving higher accuracy and lower error rates. The model's ability to handle missing data and perform implicit feature selection contributes to its popularity in healthcare applications.

Autoencoders, a type of unsupervised neural network, are commonly used for dimensionality reduction, feature learning, and anomaly detection. Hinton and Salakhutdinov (2006) demonstrated how autoencoders could effectively compress high-dimensional data, preserving essential information. In the context of heart disease, autoencoders have been employed for both unsupervised detection and pre-training classifiers. For instance, Rahhal et al. (2019) applied stacked autoencoders to extract features from electrocardiogram (ECG) data, significantly enhancing classification performance.

Several studies have suggested combining deep learning with traditional ML for improved prediction. For example, Sajjad et al. (2020) proposed a hybrid model using deep feature extraction via autoencoders followed by classification using gradient boosting, achieving higher sensitivity and specificity in medical imaging datasets. Such hybrid methods leverage the strength of deep representation and efficient classification. Accurate evaluation is critical in healthcare ML applications, where false negatives can have serious consequences. Metrics such as **Accuracy, Precision, Recall (Sensitivity), Specificity**, and **F1 Score** are essential for comprehensive model assessment. Researchers emphasize the importance of balanced performance across these metrics, especially when working with imbalanced datasets (Saito & Rehmsmeier, 2015). In this literature highlights the effectiveness of both XGBoost and Autoencoders in healthcare diagnosis tasks. However, few studies have performed a direct comparison or integration of these two models specifically for heart disease prediction using the Cardiovascular Disease Dataset. This research aims to address that gap by evaluating and enhancing prediction performance through both independent and hybrid modeling approaches.

III. RESEARCH METHODOLOGY

This section outlines the dataset used in the study, the preprocessing steps applied to prepare the data, and the machine learning models developed for heart disease prediction.

3.1 Dataset Description

The dataset used in this research is the **Cardiovascular Disease Dataset**, which is publicly available on Kaggle. It comprises anonymized patient medical records collected from routine medical examinations and is intended to predict the presence or absence of cardiovascular disease using various clinical and demographic attributes. The dataset can be accessed via Kaggle at the following URL: https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset.

The dataset contains a total of **70,000 instances** and includes **13 features**, excluding the target variable. The **target variable**, labeled *cardio*, is binary: a value of **0** indicates the absence of cardiovascular disease, while a value of **1** denotes its presence.

The features in the dataset are as follows:

- **age**: Age in days
- **gender**: 1 = women, 2 = men
- **height**: Height in centimeters
- **weight**: Weight in kilograms
- **ap_hi**: Systolic blood pressure
- **ap_lo**: Diastolic blood pressure
- **cholesterol**: 1 = normal, 2 = above normal, 3 = well above normal
- **gluc**: Glucose level (on the same scale as cholesterol)
- **smoke**: Smoking status (0 = no, 1 = yes)
- **alco**: Alcohol intake (0 = no, 1 = yes)
- **active**: Physical activity status (0 = no, 1 = yes)
- **cardio**: Target variable (0 = no cardiovascular disease, 1 = presence of disease)

This comprehensive dataset is well-suited for training and evaluating predictive models related to cardiovascular health.

Data Preprocessing Steps:

- ✓ **Handling Missing or Anomalous Values:**
  - Although the dataset contains no explicit missing values, some entries had **physiologically implausible values**, such as very high or low blood pressure, height, or weight.
  - Outliers were removed using interquartile range (IQR) filtering and domain-specific thresholds.
- ✓ **Feature Engineering:**
  - A new feature, **Body Mass Index (BMI)**, was derived using weight and height.
  - Age was converted from days to years for interpretability.
- ✓ **Encoding:**
  - Most features are numerical or already encoded.
  - The gender feature was one-hot encoded to avoid ordinal bias.
  - Categorical features like cholesterol and gluc were treated as ordinal based on severity.

- ✓ **Normalization/Scaling:**
  - Features were **standardized** (zero mean, unit variance) using StandardScaler for algorithms sensitive to feature scale, such as XGBoost and Autoencoders.
  - Normalization helped improve convergence and model stability during training.
- ✓ **Data Splitting:**
  - The dataset was split into **training (80%)** and **testing (20%)** sets using stratified sampling to maintain class balance.
  - A further **cross-validation (5-fold)** scheme was applied during training to ensure robustness.

3.2. Model Development

A.XGBoost Classifier

XGBoost (Extreme Gradient Boosting) is a powerful and scalable machine learning algorithm developed by Chen and Guestrin (2016). It is an efficient implementation of the gradient boosting framework and has become one of the most widely used algorithms for classification and regression problems, especially on structured (tabular) data. In essence, XGBoost builds an ensemble of decision trees in a sequential manner, where each new tree attempts to correct the errors made by the previous trees. The objective is to minimize a regularized loss function that improves model generalization while optimizing predictive accuracy. Why XGBoost for Heart Disease Prediction?

- Handles structured medical data well
- Effective with imbalanced classes
- Built-in regularization to reduce overfitting
- Fast training and parallel computation
- Automatic handling of missing data

XGBoost Algorithm:

XGBoost minimizes a **regularized objective function** using gradient boosting. The objective function consists of two parts:

$$L(\phi) = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t)}\right) + \sum_{k=1}^{t} \Omega(f_k)$$

Where, $l\left(y_i, \hat{y}_i^{(t)}\right)$ is a differentiable convex loss function (e.g., logistic loss). $\hat{y}_i^{(t)}$ is the prediction for the ith instance at iteration t. $\Omega(f_k)$ is the regularization term.

1. Additive Training: At each boosting round t, the model adds a new function ft(x) to improve predictions:

$$\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$$

Each ft is a decision tree from the function space F of regression trees.

2. Objective Function Approximation: To optimize the objective, we apply a second-order Taylor expansion:

$$L^{(t)} \approx \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega f_t$$

Where, $g_i = \frac{\vartheta t(y_i, \hat{y}_i^{t-1})}{\vartheta \hat{y}_i^{(t-1)^2}}$ (1st order gradient), $h_i = \frac{\vartheta t(y_i, \hat{y}_i^{t-1})}{\vartheta \hat{y}_i^{(t-1)^2}}$ (2nd order gradient)

3. Regularization Term: Regularization penalizes tree complexity,

$$\Omega(f) = \lambda^T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2$$

Where, T = number of leaves in the tree, wj = score on leaf j, γ = complexity cost for adding a new leaf AND λ = L2 regularization term on weights

4. Optimal Split (Gain Function)

To decide the best split in a tree, XGBoost computes the **Gain**:

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda} \right] - \gamma$$

Where, GL, GR = sum of gradients for left and right child and HL, HR = sum of Hessians for left and right child, A split is accepted if the gain exceeds the threshold γ.

B.Autoencoders

Autoencoders are a type of **unsupervised artificial neural network** designed to learn efficient representations (encodings) of input data, typically for the purpose of dimensionality reduction or feature extraction. Introduced by Hinton and Salakhutdinov (2006), autoencoders are especially useful in medical domains where data may be high-dimensional or noisy. In heart disease prediction, autoencoders can serve either as **standalone anomaly detectors** or as **feature extractors** that transform data into a compressed form fed into a traditional classifier (e.g., XGBoost).

Architecture: Encoder–Decoder Structure,

Autoencoders consist of two main components:

1. **Encoder**: Compresses the input x into a lower-dimensional representation z.
2. **Decoder**: Reconstructs the input from the encoded representation z to approximate the original input.

Basic Autoencoder Diagram:

Input (x) → [Encoder] → Latent space (z) → [Decoder] → Output ($\hat{x}$)

Let $x \in R^n$ be the input vector.

1. Encoder Function

The encoder maps input x to a latent representation z: $Z = f(x) = \sigma(w_e x + b_e)$

Where, W_e = weight matrix for encoder, b_e = bias vector for encoder, σ = activation function (e.g., ReLU, tanh)

2. Decoder Function

The decoder reconstructs the input from latent space: $\hat{x} = g(z) = \sigma'(w_d z + b_d)$

Where, $\hat{x}$ = reconstructed output, W_d = weight matrix for decoder, b_d = bias vector for decoder and σ′ = decoder activation function (often sigmoid)

3. Loss Function

The goal is to minimize the **reconstruction error** between the input x and the output $\hat{x}$ A common loss function is **Mean Squared Error (MSE)**:

$$L_{AE} = \frac{1}{n}\sum_{i=1}^{n} ||x_i - \hat{x}_i||^2$$

Alternatively, **binary cross-entropy** is used if data is binary or normalized between 0 and 1.

Autoencoder Variants,

- **Denoising Autoencoder (DAE)**: Learns robust features by reconstructing clean inputs from corrupted versions.
- **Sparse Autoencoder**: Adds a sparsity constraint to the latent vector to force the model to learn compact, informative features.
- **Stacked Autoencoder**: Composed of multiple layers of autoencoders, allowing for hierarchical feature extraction.

In this study, the autoencoder was used in two ways:

1. As a Standalone Model (Anomaly Detection)

The idea is that healthy (class 0) and diseased (class 1) patients may produce significantly different reconstruction errors. After training the autoencoder, we use reconstruction error as a decision boundary:

- If $L_{AE}(x) > \theta$, classify as diseased.
- Else, classify as healthy.

2. As a Feature Extractor (Hybrid Model)

The encoder part of the autoencoder was used to **extract compressed features**, which were then passed into a classifier such as **XGBoost**:

- Raw input x → Encoder → Latent vector z → Classifier

This hybrid approach often results in better performance by capturing nonlinear dependencies and reducing noise, especially in high-dimensional data like medical records.

Training Procedure

- **Activation Functions**: ReLU for encoder layers, sigmoid for the output layer (for normalized data)
- **Loss Function**: MSE
- **Optimizer**: Adam optimizer with learning rate α=0.001\alpha = 0.001α=0.001
- **Batch size**: 128
- **Epochs**: 100–200 (with early stopping based on validation loss)
- **Normalization**: Input data scaled to [0, 1] for stable training

Autoencoders offer several advantages in data processing and machine learning tasks. One of their key strengths is the ability to learn nonlinear relationships within the data, enabling more complex pattern recognition than traditional linear models. They also help reduce noise and redundancy in the input features, which enhances the quality and relevance of the data. Additionally, autoencoders can uncover latent structures that may be highly beneficial for downstream tasks such as classification. This makes them particularly valuable in scenarios where labeled data is limited, as they can effectively extract meaningful representations from unlabeled datasets.

IV.RESULT AND DISCUSSION

This section outlines the experimental design used to evaluate the predictive performance of heart disease detection models, including the environment setup, selected hyperparameters, evaluation metrics, and a comparative analysis between XGBoost and Autoencoders using the Cardiovascular Disease Dataset.

4.1 Experimental Environment

The experiments in this study were carried out on a standard computing environment with modest specifications. The system was configured with Windows 7 as the operating system, 4 GB of RAM, and a 1 TB HDD for storage. It was powered by an Intel Dual-Core processor running at 2.4 GHz. All experiments were implemented using Python 3.9. A range of libraries and tools were employed for different

tasks: scikit-learn was used for data preprocessing and model evaluation, XGBoost for model implementation, and TensorFlow/Keras for building and training autoencoders. Additionally, pandas, NumPy, matplotlib, and seaborn supported data analysis and visualization. Despite the limitations in system resources, models were successfully trained using batch processing and optimized data handling techniques.

The dataset used in the experiments is the Cardiovascular Disease Dataset, obtained from Kaggle. It comprises 70,000 instances, with 13 input features and 1 target variable. The target variable, cardio, is used for binary classification, where 0 indicates no cardiovascular disease and 1 indicates the presence of disease.

4.2 Hyperparameters

XGBoost:

Parameter	Value
learning_rate	0.1
max_depth	6
n_estimators	150
subsample	0.8
colsample_bytree	0.8
gamma	0
lambda	1
scale_pos_weight	1
objective	binary:logistic

Autoencoder:

Component	Configuration
Architecture	13 → 8 → 4 → 8 → 13 (Symmetric)
Activation	ReLU (hidden), Sigmoid (output)
Optimizer	Adam (learning rate = 0.001)
Loss Function	Mean Squared Error (MSE)
Epochs	150
Batch Size	128
Early Stopping	Yes (patience = 10)

In hybrid use, the **encoded feature vector (latent space)** from the encoder was passed into a traditional classifier (XGBoost) for final prediction.

4.3. Evaluation Metrics

The following metrics were used to evaluate and compare model performance:

- **Accuracy**: Overall correctness of the model, Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$
- **Precision**: Proportion of true positive predictions among all positive predictions , Precision = $\frac{TP}{TP+FP}$
- **Recall (Sensitivity)**: Proportion of true positives detected among all actual positives, Recall = $\frac{TP}{TP+FN}$
- **Specificity**: Proportion of true negatives correctly identified, Specificity = $\frac{TN}{TN+FP}$
- **F1 Score**: Harmonic mean of precision and recall, useful for imbalanced data, F1 Score = $\frac{2X+PrecisionxRecall}{Precision+Recall}$

Where, TP = True Positive, FP = False Positive TN = True Negative and FN = False Negative

4.4. Comparative Analysis

A comparative analysis was performed between: XGBoost Model, Autoencoder-based Classifier (Hybrid), Autoencoder as Standalone (Reconstruction Error Threshold). Each model was trained and tested on the same dataset using a stratified 80:20 train-test split. Cross-validation (5-fold) was applied to ensure robustness. The results are discussed in the Results and Discussion section with appropriate tables and visualizations to compare metrics such as F1 Score, precision, and specificity. This section evaluates and compares the performance of two models—XGBoost and Autoencoders—on the Cardiovascular Disease Dataset. The comparison is based on key classification metrics: Accuracy, Precision, Recall (Sensitivity), Specificity, and F1 Score.

Metric	XGBoost	Autoencoder + XGBoost	Autoencoder (Standalone)
Accuracy	0.871	0.884	0.841
Precision	0.853	0.861	0.829
Recall	0.865	0.878	0.812
Specificity	0.868	0.890	0.851
F1 Score	0.859	0.869	0.820

The Autoencoder + XGBoost hybrid model demonstrated superior performance compared to the standalone XGBoost model, owing to several technical advantages. First, the autoencoder's ability to reduce dimensionality and filter out noise allowed it to compress raw input into meaningful latent representations. This significantly enhanced the learning capability of the downstream classifier, XGBoost, by eliminating irrelevant or redundant features and emphasizing more discriminative patterns. Second, nonlinear feature learning enabled by activation functions like ReLU and sigmoid allowed the autoencoder to capture complex, nonlinear relationships within the medical data—patterns that traditional feature engineering might overlook. Additionally, the bottleneck architecture of the autoencoder acted as a form of regularization, compelling the network to retain only the most critical information and thereby helping to reduce overfitting. The model also demonstrated better generalization across unseen data, thanks to XGBoost being trained on robust, lower-dimensional embeddings instead of sparse, high-dimensional raw features. Furthermore, the latent space generated by the encoder improved class separation, making the classifier's decision boundaries sharper and more effective, particularly enhancing Recall and Specificity. Although XGBoost alone yielded solid results, the hybrid model consistently outperformed it across all evaluation metrics. The gains were especially notable in Recall, F1 Score, and Specificity, which are vital in medical

diagnosis scenarios where both false positives and false negatives carry serious consequences.

V. CONCLUSION

In this study, we proposed and evaluated two machine learning approaches—XGBoost and Autoencoders—for detecting heart disease using the Cardiovascular Disease Dataset. The primary objective was to enhance the predictive performance of heart disease detection models through advanced learning techniques and comprehensive evaluation metrics, including Accuracy, Precision, Recall, Specificity, and F1 Score. The experimental results revealed several key insights. First, XGBoost, known for its gradient boosting framework and high predictive capability, performed well on structured data. Second, Autoencoders, particularly when integrated with XGBoost in a hybrid model, significantly improved performance across all evaluated metrics. The hybrid model, which utilized the autoencoder for feature extraction and XGBoost for classification, achieved the highest results—88.4% accuracy, 87.8% recall, and an F1 score of 86.9%—outperforming both standalone models. The autoencoder's strength lies in its ability to learn compact, noise-reduced, and informative representations from complex medical data, contributing to improved classification accuracy and generalization. These findings highlight the potential of combining deep learning-based feature extraction with ensemble learning methods to develop high-performing clinical decision support systems.

Although the current results are promising, several directions for future research remain. First, integrating additional clinical features—such as ECG readings, family history, cholesterol levels, and imaging data—could further enhance the model's accuracy and robustness. Second, exploring advanced autoencoder variants, including Variational Autoencoders (VAEs), Denoising Autoencoders, and Sparse Autoencoders, may improve the quality and interpretability of feature extraction. Third, ensuring explainability and interpretability through explainable AI (XAI) techniques like SHAP or LIME is essential for real-world clinical adoption. Finally, the model could be deployed in real-time systems within hospital environments, enabling early-warning capabilities via real-time data streaming.

VI. REFERENCES

[1]. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

[2]. Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science, 313*(5786), 504–507. https://doi.org/10.1126/science.1127647

[3]. Chaurasia, V., & Pal, S. (2018). Early prediction of heart diseases using data mining techniques. *Carpathian Journal of Electronic and Computer Engineering, 10*(1), 14–22.

[4]. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

[5]. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30, 4765–4774.

[6]. Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., & Dean, J. (2019). A guide to deep learning in healthcare. *Nature Medicine, 25*(1), 24–29. https://doi.org/10.1038/s41591-018-0316-z

[7]. Aggarwal, C. C. (2018). Neural networks and deep learning. In *Neural Networks and Deep Learning* (pp. 75–110). Springer.

[8]. Liu, F., Zhang, D., Cheng, Y., & Liu, S. (2020). Heart disease prediction based on deep learning and hybrid feature selection. *IEEE Access, 8*, 154954–154967.

[9]. Tang, Y., & He, H. (2021). Heart disease prediction with ensemble learning and feature engineering. *Computers in Biology and Medicine, 135*, 104595. https://doi.org/10.1016/j.compbiomed.2021.104595

[10]. Kingma, D. P., & Welling, M. (2013). Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*.

[11]. Miotto, R., Wang, F., Wang, S., Jiang, X., & Dudley, J. T. (2018). Deep learning for healthcare: Review, opportunities and challenges. *Briefings in Bioinformatics, 19*(6), 1236–1246.

[12]. Huang, G. B., Zhu, Q. Y., & Siew, C. K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing, 70*(1-3), 489–501.

[13]. Chaurasia, V., & Pal, S. (2018). Data mining techniques: To predict and resolve heart disease. *International Journal of Computer Applications, 175*(10), 7–11.

[14]. Chollet, F. (2017). *Deep learning with Python*. Manning Publications.

[15]. Kaggle. (2018). Cardiovascular Disease Dataset. Retrieved from https://www.kaggle.com/sulianova/cardiovascular-disease-dataset