Comparing ML and DL classification methods for the prediction of heart disease

Mordechai Mushkin, Dmitry Strizhak, and Nathan Schumann

Abstract

The aim of the current research is to evaluate any potential advantage of Deep Learning (DL) techniques over Machine Learning (ML) techniques for predicting heart disease conditions based on tabular medical data of the patient. The research is performed over the Cleveland Heart Disease dataset taken from the UCI repository. The ML techniques addressed are basic Linear Regression (LR) and advanced Random Forest (RF) and the advanced RF and Extreme Gradient Boosting (XGBoost); and the DL technique investigated is Multi-Layer Perceptron (MLP), which is suitable for tabular data. Various MLP architectures and hyperparameters are evaluated, looking for the optimal configuration. The main outcome of the research is that the optimal MLP architecture is a single hidden layer NN of moderate size (e.g. 60 neurons), and the performance of this NN is not better than that of RF.

Keywords: ML ,DL, classification, heart disease

Introduction

Heart attacks are one of the leading causes of mortality, making early detection of heart disease a critical public health objective. The aim of the current research is to evaluate any potential advantage of Deep Learning (DL) techniques over Machine Learning (ML) techniques for predicting heart disease conditions based on tabular medical data of the patient.

The research is performed over the Cleveland Heart Disease dataset taken retrieved from the Kaggle website.

The ML techniques addressed in the research are basic Linear Regression (LR) and advanced Random Forest (RF) and the advanced RF and Extreme Gradient Boosting (XGBoost). The performance of the RF technique (0.84 accuracy) is found to be slightly better than that of LR (0.803 accuracy) and XGBoost (0.77 accuracy). Those performance results are used as the baseline for evaluation the DL techniques.

The DL techniques investigated in the research are variances of Multi-Layer Perceptron (MLP), since this technique is suitable for tabular data. Various MLP architectures, with various hyperparameters are evaluated, looking for the optimal configuration. The main outcome of the research is that the optimal MLP architecture is a single hidden layer MLP of moderate size (e.g. 60 neurons), and the performance of this MLP (0.82 accuracy) is not better than that of RF (0.84 accuracy).

Further outcome of the research is that the dataset is tight and attempting to remove potential outliers or low correlated features degrade the performances of the classifier.

Related Work

Following is a list of some papers reportion about utilizing ML techniques for the prediction of heart disease conditions.

- (Alshraideh, et al., 2024) The researchers used the JUH Heart Disease dataset from Jordan University Hospital (486 cases with 58 attributes), the study applies multiple machine learning algorithms—including SVM, RF, decision tree, naive Bayes, and KNN—with particle swarm optimization (PSO) for feature selection. Models are rigorously evaluated via 10-fold cross-validation using metrics such as accuracy, precision, recall, F1-score, and ROC AUC. Notably, SVM with PSO achieves 94.3% accuracy. The study concludes that optimized machine learning models significantly enhance early heart disease prediction, supporting timely diagnosis and improved patient outcomes.
- (Rojek, Kotlarz, w Kozielski, Jagodzi ´nski, & Królikowski, 2024) Using a patient dataset capturing diverse clinical parameters, this study develops an AI-based tool to predict heart attack risk for preventive medicine. Multiple machine learning models—Linear SVC, LR, KNN, and RF —were compared to determine personalized risk and identify a minimal feature set (heart rate, age, BMI, cholesterol). Evaluation revealed that LR, while moderately predictive, provided the most accurate results for initial screening. The system offers a rapid, cost-effective, and non-invasive predictive analysis, enabling early intervention and improved preclinical care.
- (Aghamohammadi, Madan, Ki Hong, & Watson, 2019) The paper proposes a novel classification method that combines a Genetic Algorithm (GA) with an Adaptive Neural Fuzzy Inference System (ANFIS) to predict heart attack risk using the Cleveland dataset (297 patients, 14 features). The system categorizes risk from no to

very high and is evaluated using sensitivity, specificity, precision, accuracy, RMSE, and 9-fold cross-validation. Training and testing results ireduction RMSEactory performance, with significant RMSE reduction. Overall, explainable outputs and an Importance Evaluation Function (IEF) robustly reveal key predictive features, demonstrating transparent and effective diagnosis.

Several (at least two) papers calming the utilization of various DL techniques, including CNN, RNN, LSTM, GRU and hybrid CNN-GRU architectures, over the Cleveland Heart Disease dataset , were found. However, the reliability of those papers is questionable, since none of those techniques are suitable for a tabular data-set.

Methodology

Dataset

Description of the dataset

The research is performed over the Cleveland Heart Disease dataset from the UCI Machine Learning Repository, which was retrieved from the Kaggle website.

This dataset contains information about different health and lifestyle factors that may influence heart disease in the USA. It includes details like age, cholesterol, blood pressure, and smoking habits, along with outcomes like whether heart disease is present. The goal is to help identify potential risks and trends that can lead to better heart health awareness and prevention.

The dataset contains 303 sample (patients) and 14 features. The features are presented in table 1 below.

Table 1 - The features of the dataset

Feature	Description	Values/Notes
age	Age of the patient	(years)
sex	Gender	1 = male, 0 = female
cp	Chest pain type	0: Typical angina 1: Atypical angina 2: Non-anginal pain 3: Asymptomatic
trestbps	Resting blood pressure	(mm Hg)

Feature	Description	Values/Notes
chol	Serum cholesterol	(mg/dl)
fbs	Fasting blood sugar	1 = true, 0 = false
restecg	Resting ECG results	0: Normal 1: ST-T wave abnormality 2: Left ventricular hypertrophy
thalach	Max heart rate achieved	(bpm)
exang	Exercise-induced angina	1 = yes, 0 = no
oldpeak	ST depression (exercise vs. rest)	(numeric)
slope	Slope of peak exercise ST segment	0: Upsloping 1: Flat 2: Downsloping
ca	Major vessels colored by fluoroscopy	(0–3)
thal	Thalassemia	0: Normal 1: Fixed defect 2: Reversible defect
target	Heart disease diagnosis	0: No significant disease 1: Significant disease

Preprocessing of the dataset

There was no need to clean the dataset - there are no missing values and no invalid values.

Most of the categorial features are binary.

The non-binary categorial predictors were encoded in one hot encoding.

The numeric features were scaled to the [0 1] range.

Evaluation of the characteristics of the predictor

The correlation between the features is visualized by a heat map in figure 1 below, and the correlation between the predictors are the features is presented in table 2 below.

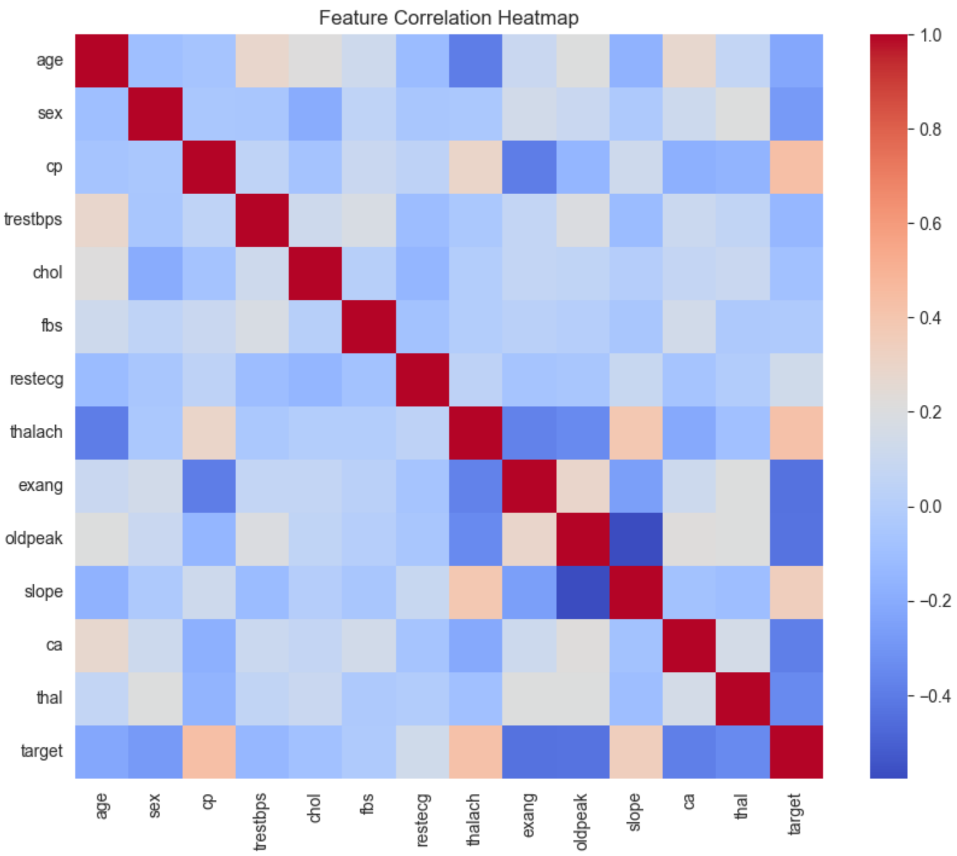

Figure 1: Feature Correlation Heatmap

Table 2 - Correlation with the target

predictor	Correlation with target
exang	-0.436757
cp	0.433798
oldpeak	-0.430696
thalach	0.421741
ca	-0.391724
slope	0.345877
thal	-0.344029
sex	-0.280937
age	-0.225439
trestbps	-0.144931
restecg	0.137230

predictor	Correlation with target
chol	-0.085239
fbs	-0.028046

The results indicate that the predictors exang, cp, oldpeak, and thalach have moderate correlation with the target, and the predictors ca, slope, thal, sex, and age have weak correlation with the target and trestbps, restecg, chol, and fbs have no correlation with the target.

Collinearity has been checked as well, and found slop-oldpeak is the one that may be considered to have almost moderate collinearity. All the rest pairs have low or acceptable collinearity.

Dataset split

The dataset is divided into a 60% training subset, 20% Validation subset, and test 20% test subset.

ML techniques

Three ML classifiers were applied to the data set: LR, RF, and XGBoost. The accuracy of these classifiers are presented in table 3 below.

Table 3 - Accuracy of the ML classifiers

Classifier	Accuracy
LR	0.803
RF	0.836
XGBC	0.77

The RF classifier achieves the best performances, and it is taken as a benchmark for the DL classifiers.

The performances of the RF classifier with respect to the accuracy, recall, precision, and F1-score are presented in table 4 below.

Table 4 The performance of the Rf classifier

Metric	Value
Accuracy	0.8361
Recall	0.9062
Precision	0.8056
F1-Score	0.8529

Applying DL techniques

Basic MLP

The first DL technique to be evaluated was an MLP with a single hidden layer of N=64 neurons. The activation function of the hidden layer was ReLU, the activation function of the output layer was sigmoid, and the loss function was Binary-Cross-Entropy (BCE).

The training process of the basic MLP is presented if figure 2 below. At the end of the training phase, the weights were restored to the epoch with the best validation loss (epoch 43 with a loss value of 0.4472). The performances of the basic MLP are presented in table 5 below.

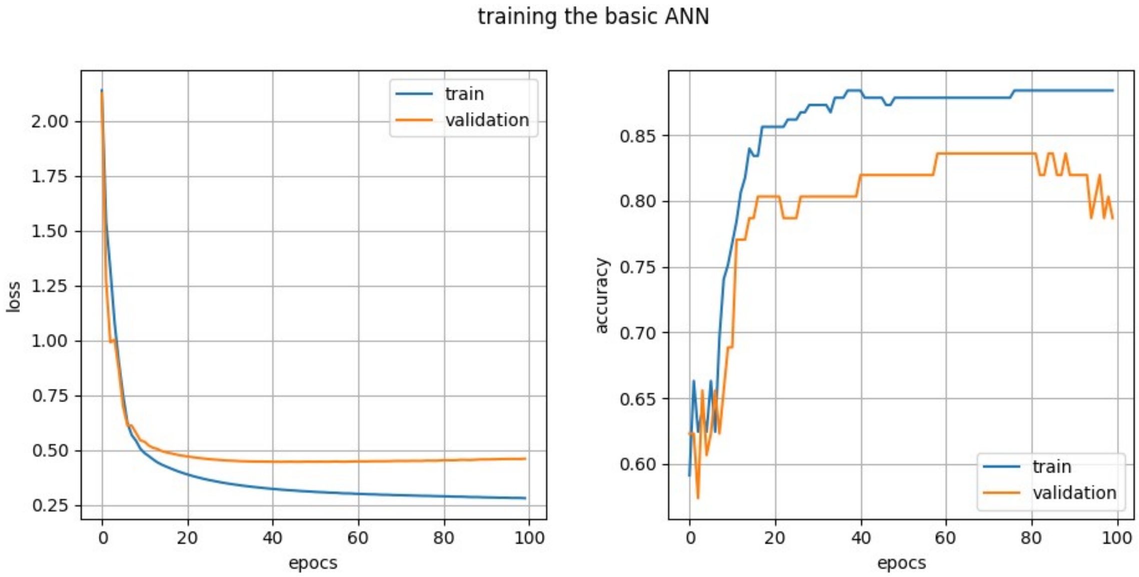

Figure 2: Basic MLP training curves

Table 5 - performances of the basic MLP

Metric	Value
Accuracy	0.82
Recall	0.84
Precision	0.82
F1-Score	0.83

Various number of hidden layers, with various numbers of neurons

Single hidden layer with various number of neurons

MLP networks containing a single hidden layer with N1 = 20, 30, 40, 60, 80, or 100 neurons were evaluated. The validation loss curves of those networks during the training process are presented in figure 3 below. Based on those curves, we have selected N1=60 as the optimal value. The performances of the optimal single hidden layer MLP are presented in table 6 below.

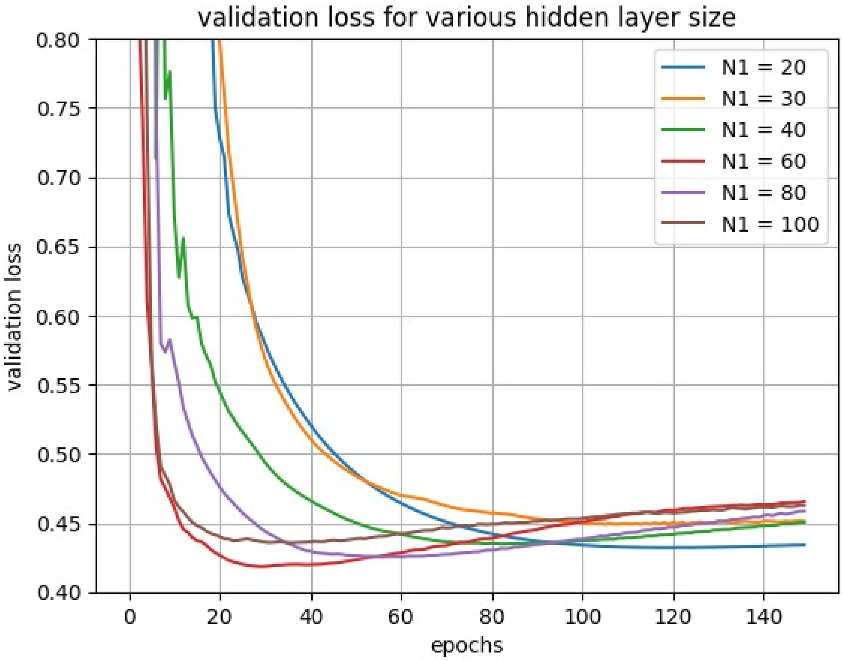

Figure 3 Validation loss curves for a single hidden layer MLP

Table 6 performances of a single hidden layer MLP (N1=60)

Metric	Value
Accuracy	0.82
Recall	0.91
Precision	0.78
F1-Score	0.84

Second hidden layer with various number of neurons

MLP networks containing two hidden layers with N1 = 60 and N2 = 4,6,8,10 or 12 neurons were evaluated. The validation loss curves of those networks during the training process are presented in figure 4 below. Based on those curves, we have selected N2=10 as the optimal value. The performances of the optimal two hidden layer MLP are the same as those of the optimal single hidden layer MLP.

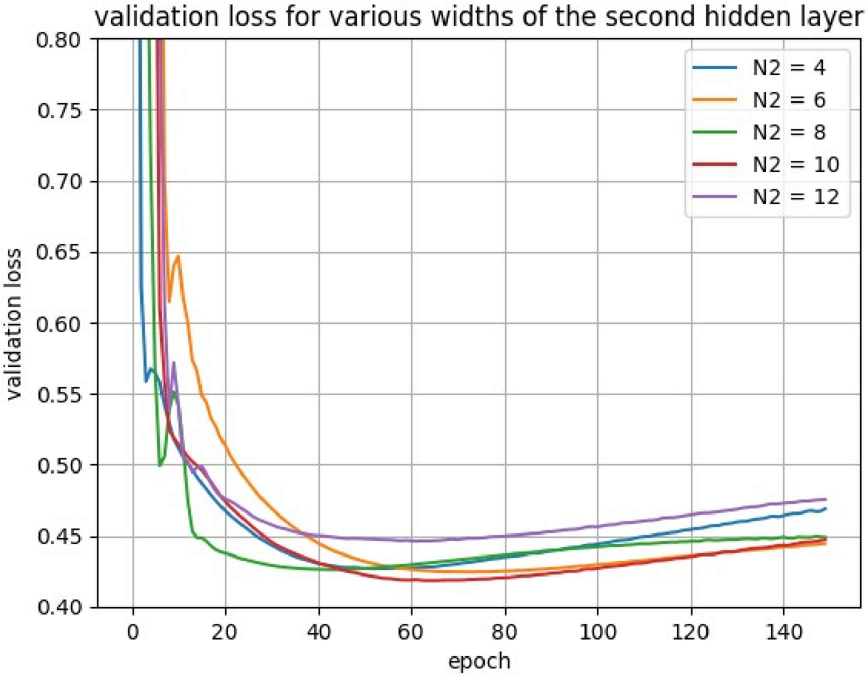

Figure 4 - validation loss curves for two hidden layers MLP

Third hidden layer with various number of neurons

MLP networks containing three hidden layers with N1 = 60 , N2 = 10 and N3 = 16, 20, 24, 28, or 32 neurons were evaluated. The validation loss curves of those networks during the training process are presented in figure 5 below. Based on those curves, we have selected N3=28 as the optimal value. The performances of the optimal three hidden layer MLP are presented in table 7 below.

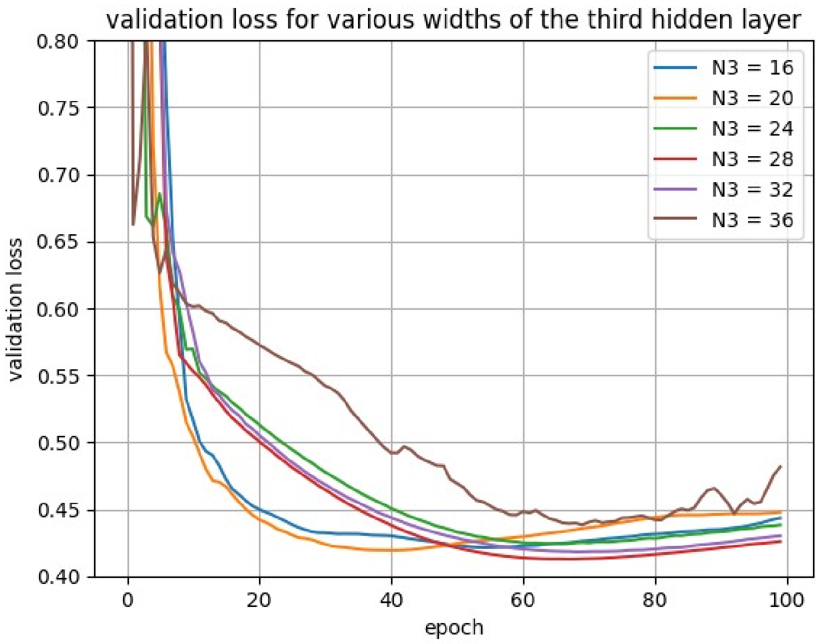

Figure 5 -Validation loss curves for three hidden layer MLP

Table 7 - Performances of the optimal three layers MLP (N3=28)

Metric	Value
Accuracy	0.80
Recall	0.91
Precision	0.76
F1-Score	0.83

As we can see, the performance of the optimal single hidden layer MLP is slightly better than those of the optimal two and three layers MPLs, while its complexity is lower. Therefore, it is selected as the optimal MLP.

A comparison of the performance of the RF classifier with that of the optimal MLP of is presented in table 8 below, showing very slight advantage to the former.

Table 8 - RF versus MLP performances

Metric	RF	MLP
Accuracy	0.84	0.82
Recall	0.91	0.91
Precision	0.81	0.78
F1-Score	0.85	0.84

Further architectural modifications

Trying to further improve the performance of the MLP classifier, we have tried to apply dropout, and to change the loss function from BCE to Hinge.

Dropout

We have applied dropout at of 0.3 at the output of the hidden layer of the optimal PLM. The loss and accuracy curves when dropout is applied are presented in figure 6 below, and the performances of the PLM with dropout is presented in table 9 below. We see that applying dropout does not affect performance.

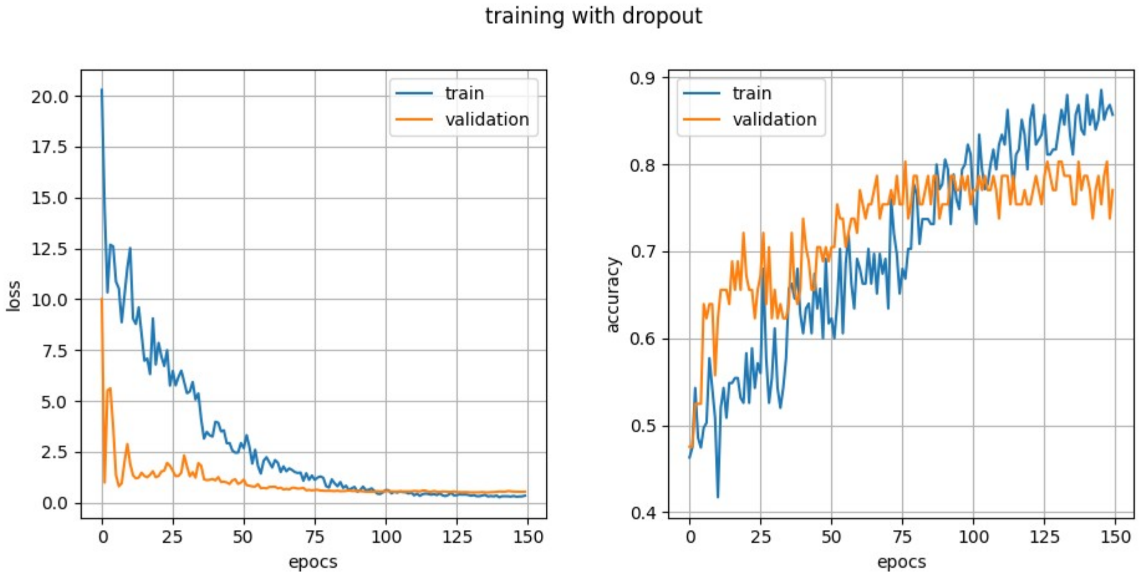

Figure 6 - Training curves with dropout

Table 9 - The effect of dropout no performance

Metric	Original	Dropout
Accuracy	0.82	0.82
Recall	0.91	0.91
Precision	0.78	0.78
F1-Score	0.84	0.84

Hinge loss function

We have replaced the BCE loss function of the optimal MLP with the Hinge loss function. The training curve while using the Hingle loss function is presented in figure 7 below, and the performances of the PLM with Hinge loss fun are presented in table 10 below. We see that utilizing Hingle rather than BCE has no effect on the performance.

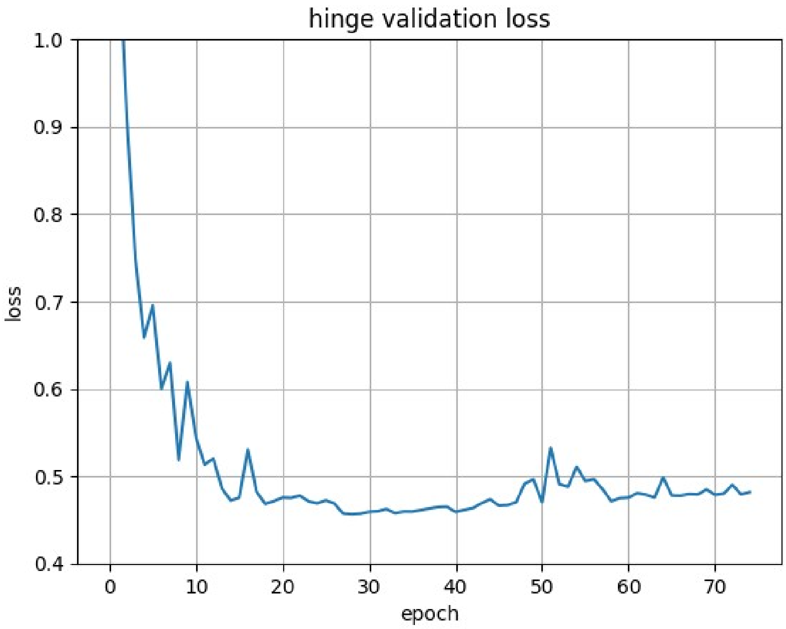

Figure 7 - Training curve with Hinge loss function

Table 10 - Hinge versus BCE performances

Metric	BCE	Hinge
Accuracy	0.82	0.82
Recall	0.91	0.91
Precision	0.78	0.78
F1-Score	0.84	0.84

Removing outliers from the data set

We have used loss per sample in order to detect outliers, remove 3% and 30% of the outliers from the training and validation sets and trained the optimal MLP on the modified sets, and applied the modified models on the test set, getting the performances results presented in table 11 below. We see that the performance was not improved but degraded in both cases.

Table 11 - The effect of removing potential outliers

Metric	Origin	3%	30%
Accuracy	0.82	0.82	0.77
Recall	0.91	0.88	0.91
Precision	0.78	0.80	0.72
F1-Score	0.84	0.84	0.81

Dimension reduction

We have tried to remove from the dataset the four predictors which have the lowest correlation with the target (trestbps, restecg, chol, and fbs). The performances of the optimal MLP classifier before and after the removal is presented in table 8 below. We see that even the removal of those low correlative predictors slightly degrade the performances.

Table 12 - The effect of removing low correlated predictors

Metric	Original	After removal
Accuracy	0.82	0.8
Recall	0.91	0.88
Precision	0.78	0.78
F1-Score	0.84	0.82

Given the dataset's limited number of predictors (13) and the relatively streamlined architecture of the neural network, there was minimal incentive to reduce dimensionality. Moreover, it was recognized that even weakly correlated predictors may offer valuable insights when considered collectively, making their removal potentially detrimental to model performance.

Results

The main outcome of the current research is that the optimal MLP classifier for the Cleveland data set contains a single hidden layer with 60 neurons, with ReLU for the hidden layer, sigmoid for the output layer, and CBE loss, and that the performances of this MLP classifier are almost good as those of a FR classifier. A comparison of the performances is presented in table 9 below.

Table 13 - Comparing the performances of RF and MLP classifiers

Metric	RF	MLP
Accuracy	0.84	0.82
Recall	0.91	0.91
Precision	0.81	0.78
F1-Score	0.85	0.84

Further outcome of the research is that the dataset is tight and attempting to remove potential outliers or low correlated features degrade the performances of the classifier. The effect of attempting to remove potential outliers is presented in table 10 and the effect of attempting to remove low correlative predictors is presented in table … below.

Conclusions

Advanced ML techniques, e.g. RF classifier, perform well on the Cleveland data set, and DL techniques introduce unnecessary complexity without improving performance.

Furthermore, the dataset is tight and balanced, and therefore there is no room for modifying it by removing samples or features. The data set is also very balanced, and there is no room for changing the balance between the positive and the negative targets by means

On the other hand, the data set is relatively small, which might place an inherent limit on the achievable accuracy. Increasing the dataset by obtaining further real samples seems to be the best way to improve the classification accuracy.

Side experiment

Changing the balance of the dataset

The data set is well balanced. The negative/positive output ratio is 0.46/0.54. Therefore, changing the balance is expected to deteriorate performance. We have created a modified data set with moderate and high imbalance by removing 50% and 90% of the positive-target samples from the training and validation sets and check the performances summarized in table 10 below.

For removal of 50% the dataset is moderately imbalanced, the recall is highly degraded, while the precision is not affected. Consequently, the accuracy and the F1 score are also degraded.

For removal of 90% the dataset is highly imbalanced, the recall is completely degraded, while the precision is not affected. Consequently, the accuracy and the F1 score are also highly degraded.

Table 14 - the effect of removing positive target samples

Metric	Original	50% removal	90% removal
Accuracy	0.82	0.77	0.64
Recall	0.91	0.78	0.44
Precision	0.78	0.78	0.78

Note: This experiment is not be considered part of the current research.

References

Aghamohammadi, M., Madan, M., Ki Hong, J., & Watson, I. (2019). Predicting Heart Attack through Explainable. EasyChair Preprint, 2093. From https://link.springer.com/chapter/10.1007/978-3-030-22741-8_45

Alshraideh, M., Alshraideh, N., Alshraideh, A., a Alkayed, Y., Al Trabsheh, Y., & Alshraideh, B. (2024). Enhancing Heart Attack Prediction with Machine Learning: A. 2024. doi:https://onlinelibrary.wiley.com/doi/10.1155/2024/5080332

Rojek, I., Kotlarz, P., w Kozielski, M., Jagodzi ´nski, M., & Królikowski, Z. (2024). Development of AI-Based Prediction of Heart Attack Risk as an. electronics. doi:https://www.mdpi.com/2079-9292/13/2/272