Høgskolen i Østfold

# MASTER’S THESIS

## PREDICTING THROMBOSIS AND BLEEDING

**Muhammad Waqas Hameed**

Master in Applied Computer Science

Faculty of Computer Sciences
Østfold University College
Halden, Norway

# Predicting Thrombosis and Bleeding

## Using Feed-Forward Neural Networks and Tabnet

Master Thesis

Muhammad Waqas Hameed

Faculty of Computer Sciences
Østfold University College
Halden
February 15, 2023

# Abstract

The performance of feed-forward neural networks (FFNN) and the TabNet architecture in the classification of thrombosis is compared in this study. This research explored how well these models performed in terms of accuracy, precision, recall, F1 score, and AUC, and how well they performed in minimizing false negatives. The models used in this comparative study were trained on the Ri-Schedule dataset for the experiment, which contains patients' personal parameters along with clinical symptoms.

Results from the experiments in our research showed that the FFNN model outperformed the TabNet model in terms of accuracy and precision. However, the TabNet model outperformed the FFNN model in terms of recall, F1 score, and AUC. These results showed that while the FFNN model is better at accurately classifying thrombosis, the TabNet model is better at identifying cases that are at risk of being misclassified as negative.

This research highlights the importance of considering multiple metrics when evaluating the performance of machine learning models in medical applications. While accuracy is important, other metrics such as recall and the F1 score are also crucial in identifying cases that may be missed by the model. Our findings suggest that the TabNet architecture may be a useful tool for addressing the challenge of minimizing false negatives in the classification of thrombosis.

**Keywords:** Machine Learning, Thrombosis Prediction, DVT Prediction, Feed-forward Neural Networks, TabNet, Model Comparison

# Acknowledgments

I would like to express my deepest gratitude to my supervisor, Dr. Lars Vidar Magnusson, for all the guidance, support and encouragement throughout my thesis writing process. I am grateful for his advice and his willingness to answer all my questions.

I extend my sincere appreciation to my family and friends for their unwavering support, motivation, and encouragement. Without them, I would not have been able to complete this project.

Finally, I would like to thank all the people who helped me in completing this project. I am grateful for their assistance and support.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Thrombosis is a disease where the blood starts clotting and blocks the veins, which can result in life-threatening situations such as stroke and heart attack [23]. This can be due to an anomaly in the hemostasis, which is a reaction from the human body to prevent bleeding and repair the injury [20]. According to the published statistics in the United States, around 900,000 people could be diagnosed with thrombosis, where 25% of cases result in sudden death with pulmonary thrombosis, and an estimated 60,000-100,000 patients die of venous thromboembolism [21].

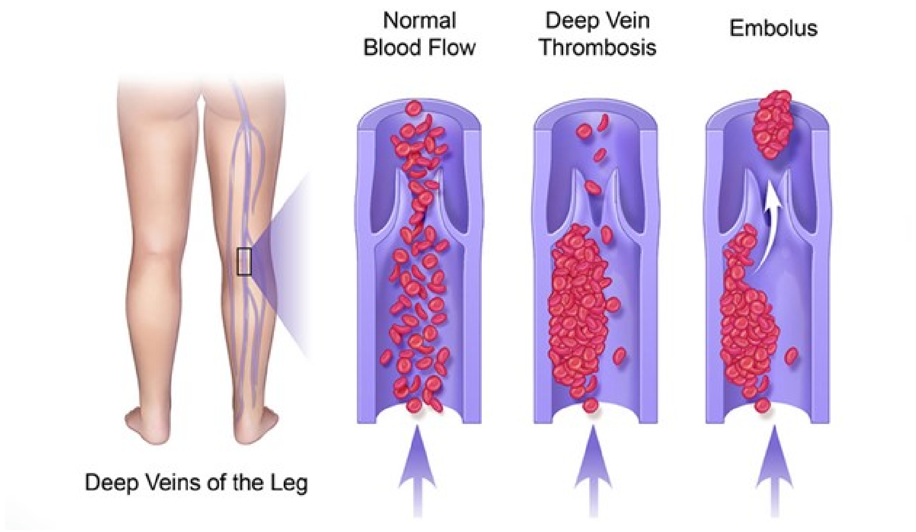


Figure 1.1: An illustration of Deep Vein Thrombosis (from www.projectpro.io)

Machine learning is revolutionizing the way healthcare operates by leveraging large amounts of data to make predictions and inform clinical decision-making. Healthcare organizations now have access to enormous volumes of patient data that can be used to train machine learning algorithms thanks to technological breakthroughs. These algorithms can then be used to identify trends and forecast the course of diseases, leading to earlier identification and more successful treatment. Healthcare practitioners can enhance patient outcomes, cut expenses, and make better decisions regarding patient care by applying machine learning in this way. Even though the application of machine learning in healthcare is still in its early stages of development, the potential advantages make it a promising field of research with significant future potential.

Figure 1.2: An example of the process of using machine learning in healthcare (from www.venousforum.org)

## 1.1 Motivation

The best defence against thrombosis is early diagnosis. Timely treatment can help prevent blood vessel scarring, and discourage future clots from emerging, and life-threatening situations [9]. Because early detection is crucial, investigating opportunities that can predict thrombosis in patients can be very useful as it can significantly reduce deaths or disabilities among the affected. In the last two decades, artificial intelligence has proven to be quite useful in many different fields to help with the predictions or decision-making, such as financial markets [24], strategic decision-making [1], autonomous vehicles [28], and most importantly healthcare [17]. Using historical data of patients, a machine learning model is trained and can be used for the prediction of various diseases [14]. Several of the techniques were explored in this research for the prediction of thrombosis using data gathered from patients.

Machine learning based various algorithms exist that can train on the provided data and learn the relations and trends between various features to predict the outcome, in this study, 'thrombosis' or 'no thrombosis'. Østfold Hospital in Halden has gathered a useful dataset containing data for the patients admitted related to the problems of thrombosis.

This research is the continuation of experimentation performed by the PhD student Ruslan Sorano [38] and MS student Khurshid Abbas [31], previously, modern machine learning algorithms, like tree-based models, have been applied by them. The goal of this study is to experiment techniques based on feed-forward neural networks and TabNet. As discussed, early diagnosis can result in tackling major life-threatening conditions and reducing the operational cost to treat the patients. Machine learning-based techniques can be the most efficient solution for prediction as they have shown convincing results in other areas of healthcare, such as medical imaging [10] and breast cancer diagnosis [18].

## 1.2 Research Objectives

This research explores the possibility of predicting thrombosis using the data gathered by Østfold Hospital. Although there are various possible machine learning techniques for the prediction, in this study, feed-forward neural networks and deep tabular data learning architecture (TabNet) will be explored and compared along with various techniques that optimize the performance of these models. The main objective of this experiment is to find which technique (used in this research) performs better and how these techniques perform in reducing false negatives while obtaining the highest accuracy. e.g., we are predicting a patient and the model predicts ‘no thrombosis’ although the patient is diagnosed with thrombosis, this type of prediction is called a ”false negative” or ”false positive.”

One of the worst scenario in the predictions are false-negative results predicted by the machine learning model, false reassurance can lead to diagnostic delay and eventually leading to slowing down the process in the treatment, therefore, it is very important that the model reduces the number of predictions for false-negatives. A detailed explanation of the experimentation setup has been included in the Experimental Setup chapter4.

**Research Question 1** How does feed-forward neural networks and TabNet architecture perform on the classification of the Ri-schedule dataset?

**Research Question 2** How do feed-forward neural networks and TabNet perform in terms of minimizing false negatives while achieving the best accuracy?

We will conduct an experimental comparison of the performance of feed-forward neural networks and TabNet architecture on a thrombosis Ri Schedule dataset to answer the research questions.

Using a set of commonly used evaluation metrics such as accuracy, precision, recall, F1 score, and AUC (PR) to compare the performance of the models. Both the models will be trained and evaluated using the same framework and on the same patients’ data. By using the same dataset and framework for both models, we will be able to ensure a fair comparison of their performance.

The experimental procedure will involve training the algorithms on the Ri-Schedule dataset and then evaluating their performance using the aforementioned metrics. We will also be using visualization techniques to present the performance of the models in a clear and easy-to-understand format. The results of the experiment will be analyzed and interpreted to provide insights on how well the models performed in terms of classification and minimizing false negatives.

This experimental approach will provide a detailed and objective comparison of the performance of FFNN and TabNet architecture on the classification of the tabular thrombosis dataset and how they perform when minimizing false negatives.

## 1.3 Report Outline

The report is structured into 7 major chapters. The ***Introduction*** chapter contains a brief description of the problem followed by motivation and research questions. The second chapter is ***Background***, which elaborates on the condition (thrombosis) along with AI techniques for predictions. Followed by the ***Related Work*** chapter, which discusses the existing research about the research topic. The ***Experimental Setup*** chapter contains an explanation of how the experimentations are conducted. The outcomes from the experiments are discussed in the ***Results*** chapter, the implications of the results are talk through in the ***Discussion*** chapter and finally, future work and conclusion are discussed in the ***Conclusion*** chapter.

# Chapter 2

# Background

This chapter is composed of the general background of a research topic. The background is supposed to exploit the summarized introduction of ”Thrombosis” and its evolution concerning literature. The contemporary and traditional techniques for diagnosing and early predicting thrombosis will be extensively discussed because of the literature review.

## 2.1 Thrombosis

Thrombosis is a multi-causal disease in medical terms and is known as ”Deep Vein Thrombosis (DVT)”. Thrombosis is considered a severe medical condition in which blood clots are developed in the blood, causing pain in the legs. Thrombosis target veins and arteries. Blood clotting disorder is the indication of the worst medical condition because clots travel through the whole human body, affecting the main organs by blocking nerves. Activated platelets may also contribute to aggravating thrombosis, increasing the inflammation in the body due to the coagulation cascade [35]. There are two major types of thrombosis, first type is Venous Thrombosis, in which blood vessels are blocked and blood is carried to the heart from the body through veins [4]. The second type is Arterial Thrombosis, in which the artery is blocked due to a blood clot. The function of the artery is to carry oxygen-rich blood far away from the heart and to the body [6]. Both types have been discussed as follows:

- **Venous Thrombosis**: A coagulation abnormality is the outcome of Venous Thrombosis in which certain factors are linked to the human body [4]. Venous Thrombosis is diagnosed in 1 out of every 1000 people, and it increases the chance of growth with age every year. The potential complications of VT are known as pulmonary embolism, this results in an abrupt mortality and a debilitating post-thrombotic syndrome. The primary causes of VT are deep vein thrombosis and pulmonary embolism. [25].
- **Arterial Thrombosis**: Atherosclerotic Plaque Rupture (Arterial Thrombosis) is regarded as a major clinical issue that persists despite ongoing advances in antithrombotic treatment [6]. There exists a general idea of seeking improvements in efficacy and safety when it comes to identifying pathogenic mechanisms that regulate arterial thrombosis. Although significant developments in this sector necessitate the early detection of potential risk factors for arterial thrombosis, The underlying causes of AT include ischemic stroke, myocardial infarction (MI), and acute limb ischemia [25].

### 2.1.1 Causes Symptoms

[27] Thrombosis is considered the third most communal cardiovascular disease after myocardial stroke and infarction [5]. Blood clots in veins and arteries are caused by thrombosis, which aggravates hypoxia, cell activation, delayed blood flow, and the release of several active compounds. There are certain causes, like recent surgery or injury, that can cause blood clots in the veins. During the period of bed rest, recovering from an ailment can increase the chance of sluggish blood flow.

There are two main symptoms of thrombosis, i.e., pain and swelling. The intensity of pain changes from cramp to acute on the affected area whereas swelling is aroused in neighbouring clots on the affected area.

### 2.1.2 Risk Factors

[27] Certain factors have been observed that contribute to causing the ailment ”thrombosis.” The factors include age, heart problems, injury or surgery, prolonged bed rest for recovery purposes, inherited genetic blood clotting disorder, hormonal imbalance, and inflammatory bowel disease.

### 2.1.3 Complications

[7] There are certain complications associated with thrombosis if it is mishandled. It can cause inflammation and acute infection, both of which can lead to hemostatic abnormalities. These abnormalities range from unworthy laboratory changes to intense increases in coagulation. This change may affect a specific part of the body, resulting in localized thrombotic complications. It is also known as systemic intravascular fibrin precipitation.

Chronic systemic inflammation activates coagulation. Due to coagulation, several complications arise, like the generation of tissue factor-mediated thrombin, the downregulation (decrease in persistent exposure) of physiological anticoagulant apparatus, and fibrinolysis suppression. Endothelium, proinflammatory cytokines, and immune cells all play important roles in fibrinolysis pathways and coagulation effects. Moreover, Indirectly or directly, the coagulation system may have a significant impact on inflammatory reactions. In terms of developing atherosclerosis and associated arterial thrombosis, the ways are likely to play a role. There are other infections than naive coagulation that can trigger thrombotic microangiopathy or hemorrhagic fever

### 2.1.4 Prevention

Thrombosis can be treated in multiple ways. for example, catheter procedures, surgeries, or medication. But the treatment of thrombosis can be instantiated with prevention through so-called ”precautionary measures.” For prevention, there are preventive medications that are recommended, such as medicines to stabilize blood pressure, blood thinners, or medications to lower cholesterol. Moreover, consuming medicines timely prescribed by a healthcare provider, regular walking to enhance movement in the body, quitting tobacco, or doing routine exercise can also prevent thrombosis. Several therapeutic surgeries may also prevent acute thrombosis [39] [26].

### 2.1.5 Diagnosis

Deep vein thrombosis (DVT) relies on objective testing as well as clinical assessment to be diagnosed. Clinical factors are considered general and might produce false-positive or false-negative results in reports [5]. The preliminary diagnosis process classifies patients into pre-labelled categories such as high, intermediate, or low-risk patients by incorporating a validated and pre-approved clinical model. Based on the ultrasound results, the category of patient is identified. Thus, thrombosis is diagnosed and confirmed after meeting the probability or threshold assigned by validating the model. In cases of low probability and normal ultrasound reports, the thrombosis is considered diminished. When the low clinical probability is merged with a negative D-dimer outcome, it also shows no thrombosis. In the absence of objective testing and clinical assessment, ultrasonography becomes vital for the diagnosis.

### 2.1.6 Treatment

Healthcare providers normally prescribe anticoagulants through IVs or injections. Catheters are used to make veins wider to dissolve the clots. Catheters are thin tubes. There is a stent that is used to make the vessel stay open and is made up of wire mesh tubes [5].

## 2.2 Machine Learning

Machine learning combined with artificial intelligence has accelerated several fields in academia, commerce, healthcare. With certain enhancements in the algorithms, deep learning is a new term that can be interchangeably used with machine learning. Also, neural networks, deep learning, and machine learning are considered sub-fields of artificial intelligence. In the healthcare sector, machine learning is known as a versatile technology. Under the umbrella of AI, machine learning is dedicated to working with labelled or structured data and performing neural network and support vector analysis. Natural language processing and deep learning have been contributing to working with unstructured data. The data is generally in the form of medical reports, demographics, physical examinations, sensor-based recordings, and so on. Machine learning combined with AI and DL uses algorithms that are designed to ”learn” features from a huge volume of data from the healthcare sector. The purpose of data extraction is to derive unknown and interesting insights that might aid clinical practices. Moreover, the accuracy and correction threshold of diagnosis and prediction of disease based on feedback can be computed. Machine learning-based AI systems are designed to minimize therapeutic and diagnostic blunders, which sometimes are inevitable in real clinical practices. Moreover, AI systems provide the latest medical information collected from cutting-edge research published by medical researchers [12].

Machine learning has evolved through four main classes. for example, supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning. These classes have been discussed in detail in sub-sections.

### 2.2.1 Supervised Learning

Machine learning makes accurate predictions and performs future outcome analysis with the help of pattern recognition, reinforcement learning, and rule-based logic. It focuses on good outcomes and discards bad ones. In the healthcare industry, samples of pathology slides are linked to training data using supervised machine learning to diagnose diseases. For example, the machine is trained with cancerous and non-cancerous cells that can decide on healthy or non-healthy images in future outcomes. After the successful identification of cancerous cells, the results are again fed into the model as test data. Supervised ML algorithms successively perform training on the input data set to build a model. Input and output labels will be distinguished by the model. Classification and regression algorithms are the two main kinds of algorithms used in supervised learning. The result of classification algorithms is a set of discrete and qualitative classes. Additionally, the output of regression algorithms uses qualitative and continuous classifications [33].

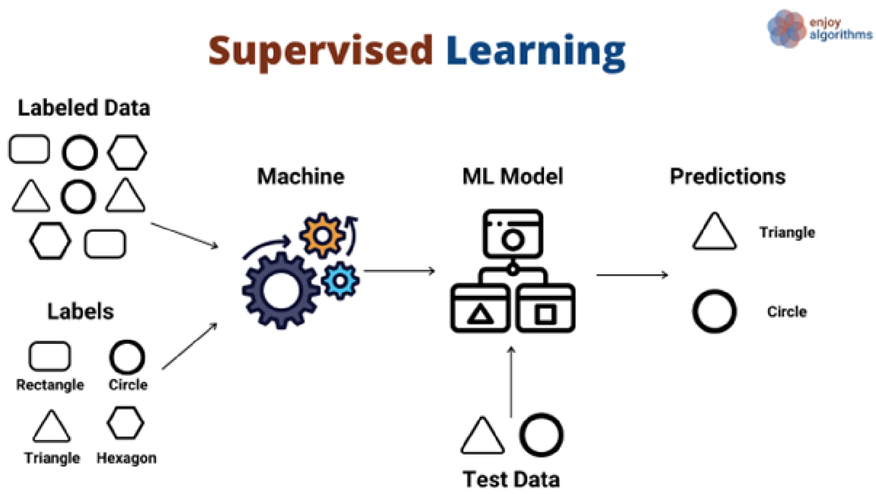


Figure 2.1: An illustration of Supervised classification learning (from enjoyalgorithms.com)

### 2.2.2 Semi-supervised Learning

Semi-supervised learning combines the features of labelled and unlabeled data, absorbing the strengths of both supervised and unsupervised data. Some data is labelled while the rest is unlabeled. For instance, there are millions of images that are given as input to the model, and some of those images are supposed to be unlabeled to deter cancer. After putting human intervention into the model, some images are labelled. Formally, the model has both types of data: labelled and unlabeled. In semi-supervised learning, unlabeled data is predicted based on labelled data every time the prediction outcome is labelled, which is called pseudo-labelling. Precisely, it uses labelled data to produce labels for more data. In healthcare, semi-supervised algorithms perform unsupervised learning on the dataset of patients after supervised learning on a small, annotated subset of data [2].

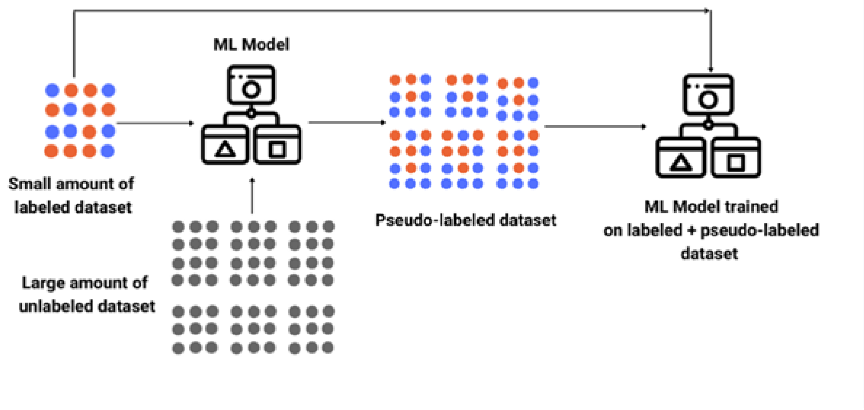


Figure 2.2: An illustration of Semi-supervised learning (from www.teksands.ai)

### 2.2.3 Unsupervised Learning

Unsupervised learning is a type that performs analysis on unlabeled data sets. In an unlabeled dataset, analysts do not predefine the expected output. For example, unsupervised ML performs dimensionality reduction and clustering. Dimensionality reduction preserves the state of the dataset while eliminating unnecessary features from the dataset, whereas clustering computes distance with neighboring nodes to determine the closest relationship between features or variables. No prior assumptions are required while implementing the algorithms. K-means clustering is the common algorithm that is used to form meaningful clusters, and principal component analysis (PCA) is a common technique to perform dimensionality reduction [33] [2].

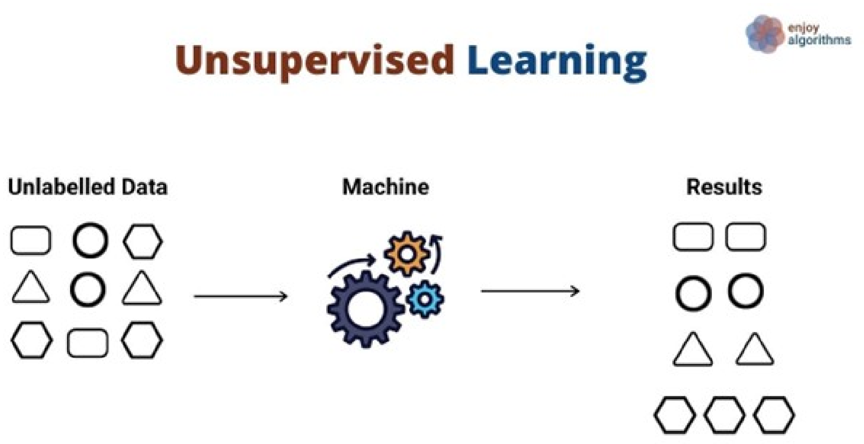


Figure 2.3: An illustration of Unsupervised learning (from enjoyalgorithms.com)

### 2.2.4 Reinforcement Learning

Policy-based machine learning (ML) is known to include reinforcement learning (RL). During the reinforcement learning process, agents (algorithms) are rewarded or penalized based on the results of their activities. RL optimizes the sequence of decisions to ascertain outcomes on a long-term basis. The result of the process is iterative improvement in which agents learn through resultant changes in the policy. RL uses features of both, supervised and unsupervised approaches. However, reinforcement learning is not currently deployed in health care because of certain limitations. The volume and non-stationarity of the data, which are partially visible, justify these constraints [33] [2].

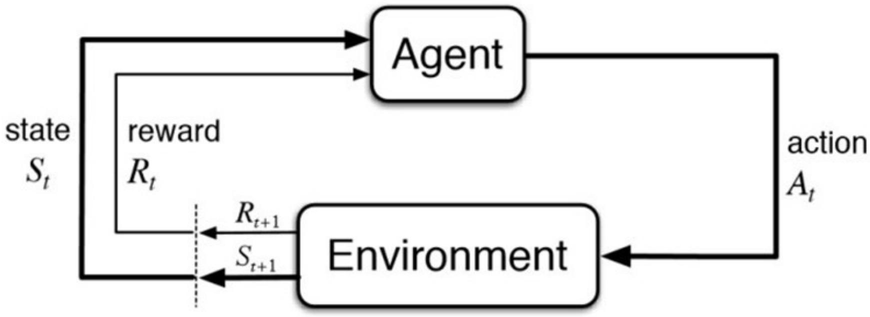


Figure 2.4: An illustration of Reinforcement learning (from kdnuggets.com)

## 2.3 Feed-forward Neural Networks

A common machine learning model for classification and regression applications is the feed-forward neural network. In this algorithm, information is processed and transmitted via a network of interconnected neurons, or nodes. Each neuron in the network gets information from other neurons, transforms it nonlinearly, and then transmits the results to other neurons in the layer below it. This process continues until the final layer of the network, where the output is generated.

The capacity of feed-forward neural networks to learn from and adjust to new data is one of their main advantages. To reduce the error between the expected output and the actual label during training, the model modifies the weights and biases of the neurons. As a result, the model can better anticipate outcomes for data that has not yet been observed and capture complex patterns and relationships in the data.

From the programmer's perspective, the programmer often over-parametrizes neural networks by adding multiple hidden layers. This act can dominate the performance of Feed-Forward NNs (FFNN). This complexity can be regulated by a hyperparameter called "weight decay." Weight decay is used to penalize the weights assigned in hidden layers. So, the complexity can be balanced by using the weight decay parameter as the hyperparameter. This hyperparameter contributes to performing grid search in FFNN for optimal setting by cross-sample validation [8].

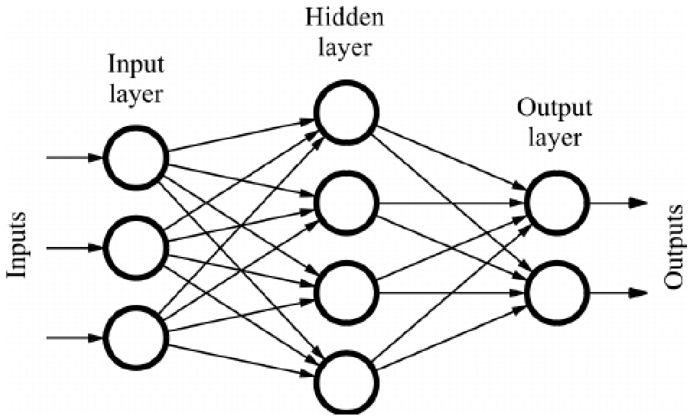


Figure 2.5: An illustration of Feed-forward Neural Networks (from deepai.org)

## 2.4 TabNet

Feed-forward neural networks have been acknowledged as performing well with images, textual, or audio/video types of data. There are canonical architectures that reside under the shelter of FFNN, which transforms raw data into meaningful information. Other than the afore-mentioned data types, a tabular form of data is yet to be processed by these canonical architectures. The tabular type of data has been explored by ensemble decision trees because decision trees have hyperplane boundaries of tabular data and represent them efficiently. Decision node tracking is quite easy in DT-based structures. Moreover, training of tabular data is quite fast in DT. In comparison with FFNN, there are some shortcomings in dealing with the tabular type of data, such as over-parameterization. In a deep learning context, the use of a huge volume of data is an assumption to achieve efficient performance. FFNN do end-to-end learning, enabling gradient descent instead of training performed by decision trees. There are certain properties of gradient descent-based learning, such as heterogeneous types of data, i.e., images with tabular data that can be encoded efficiently, feature engineering, learning streaming data, and end-to-end learning for domain adaptation for models [37].

From the architectural perspective, Tabnet is designed to input raw data without prior data processing. Data is trained via gradient descent-based optimization, and models are integrated smoothly in an end-to-end fashion. Tabnet performs feature selection based on a single instance, in which each instance is treated as a distinct one. Tabnet also provides local and global interpretability. Local interpretability refers to visualizing and combining important features, whereas global interpretability quantifies the role of each feature in the training model [32].

### 2.4.1 TabNet for Tabular Data Processing

Like conventional decision trees, a deep neural network mimics the process of decision trees with distinctive functionalities, e.g., output manifolds. Those functionalities are incorporated in the DNN design for tabular data. DNN design involves feature selection, which is considered key to acquiring decision boundaries over hyperplane. The decision boundaries are then generalized to linear feature selection combinations. In linear combinations,

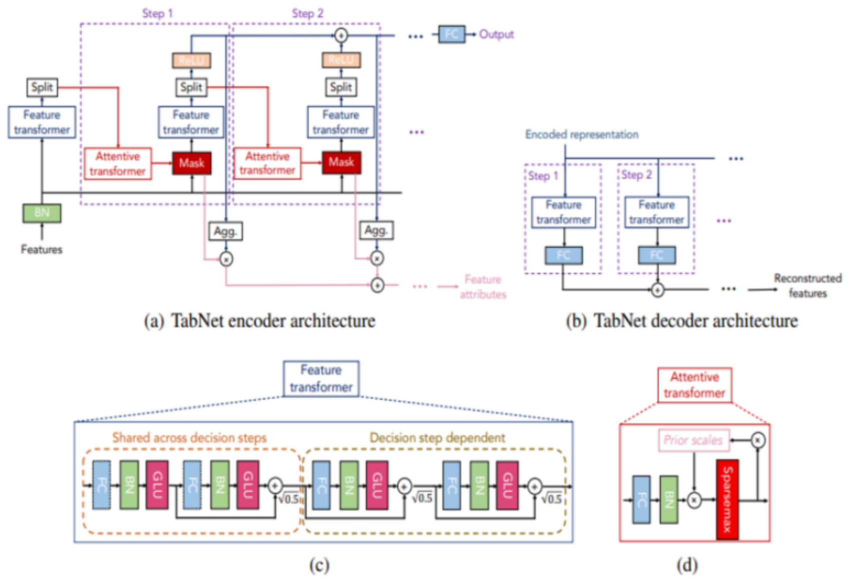


Figure 2.6: An illustration of TabNet architecture (from medium.com)

coefficients are used to determine the proportion of every single feature. The Tabnet processes feature in the same way, which makes it outperform the conventional decision tree-based approach [32]. The Tabnet design is composed of the following four assumptions:

### Feature Selection

Feature selection is performed using a combination of self-attention and gradient-based optimization.

The model can learn to give various features different weights dependent on how important they are for the job at hand thanks to self-attention processes. The most essential features for making predictions may be found using these weights, which are acquired during training.

The weights of the model are modified using gradient-based optimization based on the gradient of the loss function with respect to the model's parameters. This enables the model to discover the attributes that are most helpful for minimizing loss and enhancing performance.

Together, these processes enable TabNet to automatically identify the most pertinent features for a particular classification task and to make predictions using only those features. This can minimize the possibility of overfitting and enhance generality performance.

### Feature Processing

Feature processing is carried out by a series of layers that operate on the input data to extract useful features and representations. These layers can include a combination of convolutional, pooling, and fully connected layers, as well as self-attention mechanisms.

Local features, such as edges, corners, and patterns, are extracted from the input data using convolutional layers. The feature maps created by the convolutional layers are down-sampled using pooling layers, which lowers the dimensionality of the data and makes it easier for the rest of the network to process.

To discover non-linear correlations between the input data and the output labels, fully linked layers are used. In order to provide a more abstract representation of the input, they can also be utilized to merge the features retrieved by the convolutional and pooling layers.

The model can learn to give various features different weights dependent on how important they are for the job at hand due to self-attention processes. These weights are learned through the training process, and they can be used to identify the most relevant features for making predictions.

The feature processing in TabNet is designed to extract useful features and representations from the input data, which are then used to make predictions on the classification task.

#### Interpretability

The term "interpretability" describes a machine learning model's capacity to offer justifications for its conclusions and forecasts. This is crucial for many applications because it enables users to comprehend how the model generates predictions and to see any biases or errors that may exist.

TabNet is a deep learning architecture that has been demonstrated to produce state-of-the-art results on a range of tabular data classification problems. It is meant to perform well on classification tasks. But like many deep learning models, it can be difficult to decipher TabNet's underlying workings and determine precisely how it generates its predictions.

TabNet uses techniques such as feature importance or feature attribution methods to understand which features are most important for making predictions. These techniques can provide insight into which features are driving the model's decisions and how they are being used in the prediction process.

## 2.5 Performance Metrics

To evaluate a model's performance, evaluation measures are employed. Evaluation metrics offer a mechanism to objectively contrast a model's predictions with the dataset's actual values, and they can aid in determining the model's advantages and disadvantages.

Utilizing evaluation metrics is crucial for the training of the model and it enables us to fairly and accurately evaluate the model's performance and compare it with that of other models. We may better understand the strengths and limitations of the model and pinpoint opportunities for improvement by calculating and comparing evaluation metrics. Additionally, evaluation metrics can help us compare different models and select the best one for a specific task and dataset.

Accuracy is the proportion of correct predictions the model makes. It is determined by dividing the total number of predictions by the proportion of accurate predictions.

Precision is the proportion of reliable positive predictions that the model generates. It is calculated by dividing the overall number of positive forecasts by the actual number of positive forecasts.

Recall is the proportion of actual positive cases that the model correctly predicted. It is figured out by dividing the total number of genuine positive forecasts by the total number of real positive cases.

A metric that combines recall and precision is the The **F1 score**. It is computed as the harmonic mean of recall and precision and is frequently employed as a performance summary indicator for models.

A binary classification model's performance is assessed using the **area-under the curve - precision recall (AUC PR)** metric. It is calculated by visualizing the precision and recall for various thresholds, then calculating the precision-recall curve's area under the curve. Because it considers both precision and recall, AUC PR is a helpful indicator when there is a class imbalance in the data.

Accuracy, precision, recall, F1 score, and AUC PR are all metrics that are commonly used to evaluate the performance of a model. They provide different information about the model's performance, and can be used together to get a more comprehensive view of how well the model is able to make predictions.

# Chapter 3

# Related Work

This chapter contains a detailed literature survey in terms of the usage of machine learning in healthcare and for the prediction of thrombosis.

## 3.1 Research Paper Selection

The related research articles to the idea have been selected from diversified databases like IEEE Explore, Springer, and Science Direct. These databases are indexed by the renowned search engine Google Scholar. The relevant papers have been collected from Google Scholar and scrutinized based on predefined parameters for selection such as search strings, journal-based papers, conference-based papers, and subject-related books. In this study, patents, short papers, and Web articles have been ignored.

## 3.2 Machine Learning based Prediction of Thrombosis

This section discusses various studies on the prediction of thrombosis using various machine learning-based techniques. The structure of the related research paper discussion is described in table 3.1.

| | |
|---|---|
| 1 | Who conducted the study and what is the subject |
| 2 | Objective of the research |
| 3 | Description related to the subject of research |
| 4 | Which dataset was used and how the data was gathered |
| 5 | How did they process the data and selected the relevant features |
| 6 | Which machine learning techniques were used and their performance |
| 7 | Which evaluation metrics did they use and what were the results they achieved |
| 8 | Provide relevant illustration that describes the model/result |

Table 3.1: Structure of relevant research papers discussion.

### 3.2.1 ANNs for prediction of recurrent venous thromboembolism

A study conducted by T.D. Martins et al. [29] evaluated the performance of Artificial Neural Networks (ANN) in predicting Recurrent Venous Thromboembolism (RVTE). RVTE is a complex condition with incidence ranging from 13% to 25% in the five years following the initial episode. The objective of the study was to develop three ANN models to distinguish between the patients with the probability of exhibiting RVTE based purely on the clinical data as an alternative procedure.

The data used for the training of numerous ANN structures consist of 39 clinical factors retrieved from 235 patients. The input was the only difference midst the three models used, all 39 factors were used as input the first model, in the second model, only 18 components which were identified as major predictors of RVTE through Principal Component Analysis (PCA) were used, and in the third ANN model, practical aspects were also considered along with PCA results and a total of 15 features were used.

Three optimization algorithms were examined, each with a different number of hidden layers and neurons. All three models were used with 5-fold cross-validation. All three models included in the examination can predict RVTE, and each of the optimization techniques produces various outcomes in terms of accuracy score. The best model had a high level of accuracy, and the best structure for each of the models is illustrated in figure 1. The findings of the cross-validation demonstrated that they are consistent. The study concluded that for all sets of input parameters, ANNs are effective at predicting RVTE. The best models had a high level of accuracy, and 5-fold cross-validation verified their ability to forecast.

| Confusion Matrix for ANNs | | | | | | |
|---|---|---|---|---|---|---|
| **ANN** | **Training** | | **Validation** | | **Test** | |
| **39-10-10-1** | RVTE | Non-RVTE | RVTE | Non-RVTE | RVTE | Non-RVTE |
| **RVTE** | 32 | 7 | 6 | 0 | 6 | 3 |
| **Non-RVTE** | 4 | 122 | 1 | 28 | 1 | 25 |
| | | | | | | |
| **18-10-5-1** | RVTE | Non-RVTE | RVTE | Non-RVTE | RVTE | Non-RVTE |
| **RVTE** | 28 | 3 | 10 | 0 | 7 | 0 |
| **Non-RVTE** | 3 | 131 | 0 | 25 | 1 | 27 |
| | | | | | | |
| **15-15-10-1** | RVTE | Non-RVTE | RVTE | Non-RVTE | RVTE | Non-RVTE |
| **RVTE** | 30 | 2 | 6 | 0 | 11 | 1 |
| **Non-RVTE** | 1 | 132 | 1 | 28 | 0 | 23 |

Table 3.2: Confusion matrix with the best structure of 1,2 and 3 ANNs (source: T.D. Martins et al.)

### 3.2.2 Machine learning approaches for risk assessment of peripherally inserted Central catheter-related vein thrombosis in hospitalized patients with cancer

In order to evaluate the risk of peripherally inserted central venous catheter (PICC) related thrombosis using various machine learning approaches, S. Liu et al. [22] conducted a study for hospitalized cancer patients. PICC is a technique for directly administering medications and other treatments to the large main veins of the heart. PICC, on the other hand, has a number of shortcomings. Venous thrombosis is among the most frequent and important effects of PICC installation. It is typical in cancer patients who need long-term treatment and supportive care. In PICC-related thrombosis, the majority of clinical signs do not manifest till the thrombus has developed. Because of this, the patient will develop post-embolization syndrome, which will hinder therapy, increase discomfort and financial load, and could even result in an embolism, endangering their life.

The research included 348 cancer patients who had PICC as its source of data. For 30 days, every patient had examinations. Every week, continuous color Doppler ultrasonography was performed on the patients as a method of detecting symptomatic or asymptomatic PICC-related thrombosis. The most recent follow-up visit or PICC-related thrombosis was utilized to determine the time to event from the PICC's implantation to the thrombosis or most recent follow-up visit. The study's data set includes clinical characteristics, the onset, the course, and the outcome of thrombosis. Data from the patients were divided into training and test sets at random, with 50 percent going to each set.

For the experiment, five models—Seeley, Seeley-Random Forest (RF), Seeley-Least Absolute Shrinkage and Selection Operator (LASSO)-RF, RF, and LASSO-RF—were used and compared. Seeley is making reference to a set of standards that are frequently employed to calculate the likelihood of PICC-related thrombosis. As seen in figure "ref"fig:sliu," the results from the used models demonstrated that, as compared to a Seeley score of 0.5, all ML models performed better in terms of AUC, with scores of 0.7733, 0.7869, 0.7833, and 0.7717 as shown in Figure 3.2.2. To evaluate the reliability of the performance for each model, independent 100 repeated trials were conducted. The study found that ML approaches outperform the accepted standards for identifying cancer patients at high risk for PICC-related thrombosis.

### 3.2.3 Machine learning to predict venous thrombosis in acutely ill medical patients

Machine learning was utilized to predict venous thrombosis in a ABC study conducted by T. Nafee et al. [30] on medical patients. Acutely sick patients who are at high risk for venous thromboembolism (VTE) can be recognized clinically or with integer-based scoring systems. These outcomes demonstrated just fair performance in external data sets. The goal of the study was to compare the performance of various machine learning models with the International Medical Prevention Registry's Venous Thromboembolism (IMPROVE) score. A graded thrombosis risk assessment called the IMPROVE score was created specifically for patients with serious illnesses.

A total of 7513 hospitalized patients data was collected who have involved in phase 3 clinical trials for the experiment. The data included patients' historical medical illnesses personal details.

Predictive modeling was performed, and a total of 39 potential machine learning algorithms in 5 model families were developed using 10-fold cross-validation. Generalized additive models, elastic net (penalized logistic regression), extreme gradient boosting, random forests, a Bayesian logistic regression with default priors, and a straightforward classification tree were among the potential models developed. The super learning ensemble strategy then

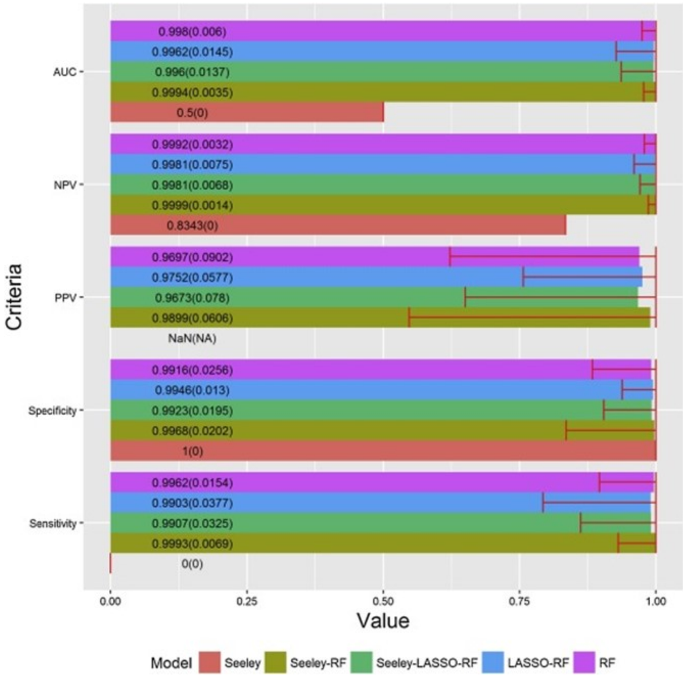


Figure 3.1: Performance comparison for each machine learning model to predict PICC-related thrombosis (source: S. Liu et al.)

used cross-validation to choose the weights to combine the outputs of each candidate classifier for more precise predictions. Two super learners were generated, the first model (ML) using all the characteristics included in the acquired dataset, and the second model (rML) using reduced variables that were thought to be key risk factors, to compare with the IMPROVE score.

According to the findings, the c-statistic scores for ML, rML, and IMPROVE were 0.69, 0.68, and 0.59, respectively. According to the study, super learners powered by machine learning are much more effective than IMPROVE at predicting thrombosis in critically unwell patients.

### 3.2.4 A Machine Learning Approach to Predict Deep Venous Thrombosis Among Hospitalized Patients

L. Ryan et al. [34] conducted a study for hospitalized patients to predict deep venous thrombosis (DVT). DVT is a serious disease which can lead to cardiovascular disease and is linked to a higher risk of sickness, death, and cost of healthcare. Risk prediction for DVT using standard DVT identification of risk factors scoring systems frequently fail to adequately classify hospitalized patients and are unable to reliably identify which inpatients are most likely to develop DVT. The objective of the study was to build machine learning models that could predict the risk of developing DVT among hospitalized patients using historical patient data collected from multiple sources.

A total of 99,237 patients' data was collected from an academic medical center's electronic medical records and was randomly split into an 80-20 proportion of training and test data,

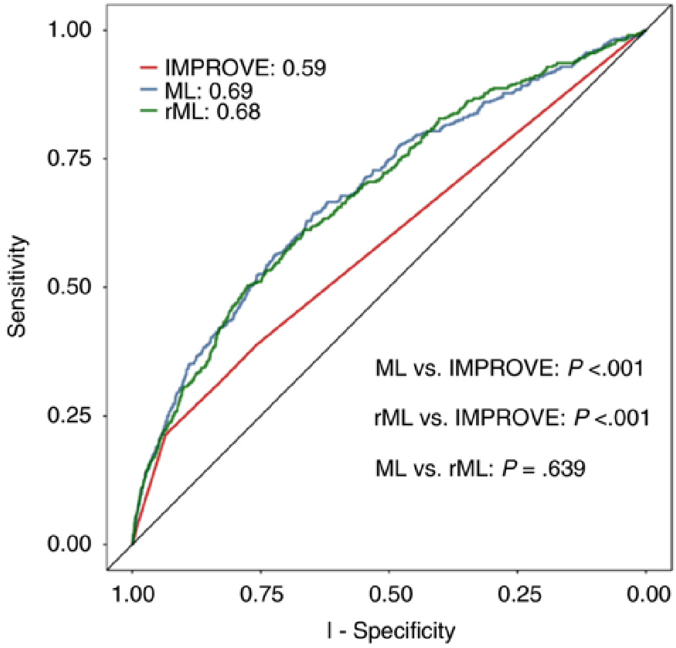


Figure 3.2: C-statistics score for machine learning based super learners compared to IMPROVE (source: T.N. MD et al.)

respectively. Patient demographics, diagnoses, vital signs, and laboratory data were all included in the data.

Two machine learning models were developed for the classification, for DVT prediction at 12 and 24 hours prior to the onset using XGBoost with 5-fold cross-validation. An intense hyperparameter optimization was performed using grid search. The results produced by the models were compared to the IMPROVE score, where across both time periods, the machine learning models outperformed the IMPROVE score. The performance metric used was the area under the receiver operating characteristic (AUROC). The scores were 0.83, 0.78, and 0.85, 0.79 for 12 hours before onset, and 24 hours before onset for machine learning models and IMPROVE, respectively, as shown in figure 4. According to the findings of the study, machine learning approaches can be very useful in predicting the risk of DVT for both 12 and 24-hour onsets.

### 3.2.5 Predicting risk for portal vein thrombosis in acute pancreatitis patients: A comparison of radial basis function artificial neural network and logistic regression models

In a study conducted Y. Fei et al. [11], the effectiveness of logistic regression models and artificial neural networks for predicting the likelihood of portal vein thrombosis in pancreatitis patients was compared. The study's goal was to create an artificial neural network (ANN) model utilizing the radial basis function (RBF) to forecast the occurrence of portal vein thrombosis brought on by acute pancreatitis (AP) (PVT). The pancreas as well as other organs are affected by AP, which is a frequent and reoccurring disorder. Portal vein thrombosis is one of the vascular effects of acute pancreatitis (PVT).

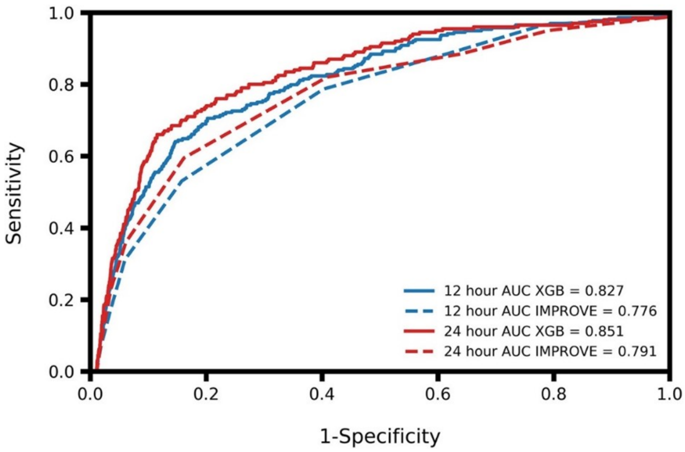


Figure 3.3: Receiver operating characteristic (ROC) curves of the XGBoost (XGB) and IMPROVE models for 12 and 24-hour prediction of DVT (source: L. Ryan et al.)

The research utilized data for 353 patients with AP. The radical basis function (RBF) artificial neural network (ANN) model and the logistic regression model were built using eleven factors that are relevant to AP. The value of the prediction in the two models was assessed using statistical indexes.

The research included data from 353 patients with acute pancreatitis (AP). The study used an artificial neural network (ANN) model based on radical basis functions (RBFs) and a logistic regression model to predict the outcome of AP. The models were built using eleven factors that are known to be relevant to AP. The performance of the models was evaluated using statistical indexes.

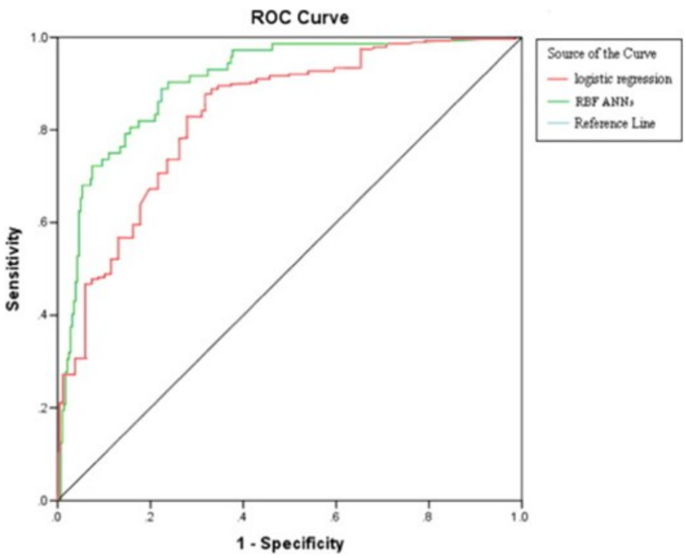


Figure 3.4: ROC for the RBF ANN's and logistic regression models (source: Y. Fei et al.)

## 3.3 Machine Learning Algorithms Comparative Studies

A machine learning framework was utilized in a study by V. Danilatou et al. [36] to predict mortality in patients with venous thromboembolism (VTE). The machine learning framework utilized in this experiment is known as JADBIO AutoML; it is a classification machine learning pipeline that makes use of an Artificial Intelligence (AI) Decision Support System called Algorithm and Hyper-Parameter Space Selection (AHPS). Linear, Ridge and Lasso Regression (LR), Decision Trees (DT), Random Forests (RF), and Support Vector Machines are the models utilized in this framework (SVMs). A series of machine learning configurations, including algorithms and hyper-parameters, are first created by JADBIO. A bootstrap-corrected cross-validation approach is then used to evaluate the configurations. When choosing the ideal configuration, JADBIO takes into account a number of statistical performance indicators, including the truth table, AUC, sensitivity, specificity, and precision as well as a few chosen features' prediction scores.

Similarly, supervised learning algorithms for RF-based diagnosis of breast cancer were compared by S. Nayak et al. [13] Decision Trees (DT), Support Vector Machines (SVM), Nave Bayes, and neural networks were among the techniques used in the study (NN). The performance was evaluated according to forecast accuracy. SVM produced the best outcomes for the detection of breast cancer among the models examined.

Another work by R. Saravanan et al. [16] analyzed supervised learning techniques that are widely used in the categorization of data. To acquire an overview of supervised ML approaches for data classification, the study contrasted the purpose, methodology, advantages, and disadvantages of the models. Additionally, supervised techniques can be further divided into probabilistic and linear classifiers. The probabilistic classifiers used in this study are the Naive Bayes Classifier (NB), Bayesian Network (BN), and Maximum Entropy Classifier (ME), whereas the linear classifiers include the Support Vector Machine (SVM), Multi-Layer Perceptron (MLP), Logistic Regression (LR), Rule-based classifiers, Decision Tree (DT), and Neural Network, as well as other According to the study's findings, supervised learning techniques are easier to employ and excel on high-quality datasets.

R. Caruana et al. [3] performed a pragmatic comparison of ten supervised learning techniques: Support Vector Machines (SVMs), Decision Trees (DT), Random Forest (RF), Logistic Regression (LR), Naïve Bayes (NB), Neural Nets (NN), memory-based learning, bagged trees, boosted trees, and boosted stumps. The impact on performance was examined for each of the models by calibrating the hyperparameters with the help of Platt Scaling and Isotonic Regression. The study used a number of performance criteria to evaluate the learning processes, such as accuracy, lift, receiver operating characteristic, average precision, cross-entropy, and root mean square error. The study concluded that calibration using either of the mentioned methods had a significant effect on obtaining the best performance for boosted trees, SVMs, boosted stums and NB whereas RF performance was also improved noticeably, the rest of the models were not significantly improved by calibration. In the experiment, overall, tuned boosted trees showed the best performance followed by untuned bagged trees, tuned SVMs, and untuned NNs, on the other hand, NB, LR, DT, and boosted stumps performed poor, respectively.

A study performed by F.Y. Osisanwo et al. [15] described, compared, and determines the best-performing classification model based on the data set, number of instances and features. The analysis was performed on the diabetes data set having 786 records, 8 features and a label. The experiment considered seven different machine learning models: Decision Table, Random Forest (RF), Naïve Bayes (NB), Support Vector Machine (SVM), Neural Network (Perceptron), JRip and Decision Tree (J48) using Waikato Environment for Knowledge Analysis (WEKA) machine learning tool. The experiment demonstrated that SVM has the best performance in terms of precision and accuracy followed by NB and RF, respectively on the diabetes data set

From the comparative study, it was noticed that for each of the problem or data sets different supervised machine learning algorithms have the best performances, so, for each of the data sets it is always the best approach to use as many algorithms as possible for the comparison along with an intense hyperparameter tuning as tuned models give best results.

# Chapter 4

# Experimental Setup

This chapter describes the setup used for the experimentation to predict thrombosis and evaluation of the classification techniques used. 4.1 describes the process followed in this research.

## 4.1 Experiment Flow

The figure 4.1 shows the process followed for the experimentation. The experimental setup was designed to evaluate the performance of a machine-learning model. First, the dataset was collected and studied to gain insights into the data. This was followed by exploratory data analysis and visualizations to better understand the data. After that, the data was split into a train and test set.

A 5-fold cross-validation method was employed to use the train sample to discover the model's optimal hyperparameters. The model was trained on the train set and assessed on the validation set during this phase. The best model was then chosen, and predictions were made on the test set using that model.

The performance of the model was evaluated and compared with the baseline model. This process was repeated for different models and the best model was selected for further evaluation.

The experimental setup was designed to select the best model and evaluate its performance on the test set. The results from this setup were used to compare the FFNN and TabNet models and draw conclusions on their performance.

## 4.2 Dataset Description

The Ri-Schedule dataset is a collection of digital clinical data obtained from the Østfold Hospital Trust in Halden, Norway, as part of the Ri-Schedule study. The study was conducted between February 2015 and November 2018 and included patients with suspected DVT who were referred to the hospital's emergency room. Each record in the original

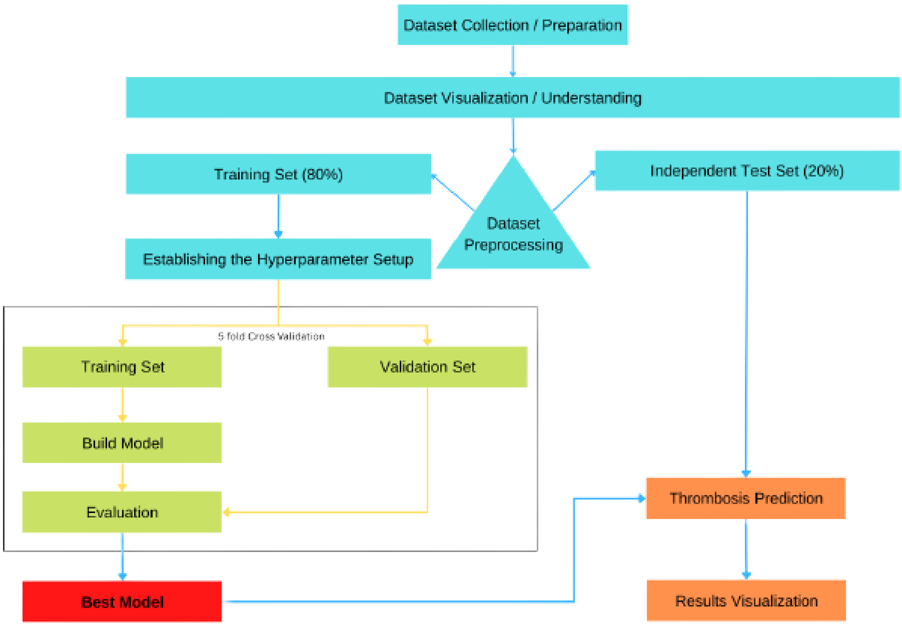


Figure 4.1: Process followed for the the selection of best model and prediction of thrombosis

dataset comprises 195 attributes and contains information on 1800 patients [38].

The features of the dataset include personal parameters such as age, gender, weight, and height, as well as clinical symptoms, risk factors, vital signs, laboratory test results, measures of the limbs, follow-up visit details, prescribed medications, and diagnostic results [38].

The labels in the dataset are the clinic nurse's assessment and the medical doctor's assessment, each of which indicates whether the patient has DVT or not. These labels were obtained through careful examination and assessment by trained professionals, and they provide reliable and accurate ground truth for evaluating the performance of the machine learning model [38].

After the necessary preprocessing has been completed for predicting DVT based on the variables that are currently accessible, the Ri-Schedule dataset provided an extensive and detailed collection of data regarding patients who have been diagnosed with suspected DVT. It can be used to train and test machine learning models in our experiments.

## 4.3 Exploratory Data Analysis (EDA)

Exploratory data analysis (EDA) is a process used to explore and understand a dataset. EDA typically involves a combination of visual and statistical methods to help better understand the data and identify patterns and relationships.

The values of the box plot in Figure 4.3 show that the age of individuals with a vte value of 1 or 0 is spread between 53 and 73, with very few records with ages less than 20

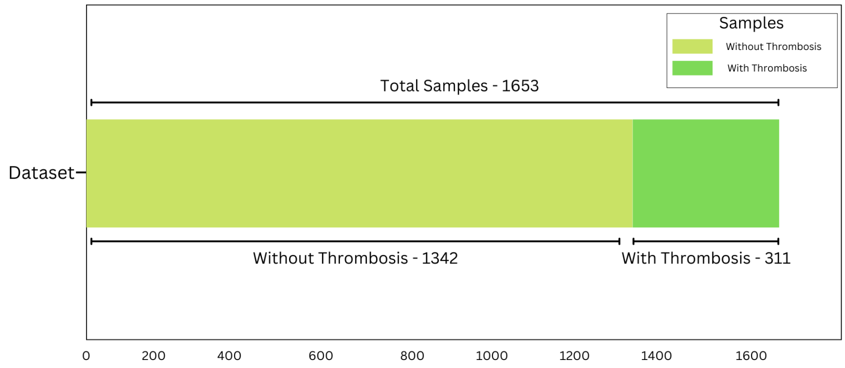


Figure 4.2: RI-Schedule study dataset description

for ”without thrombosis.” This suggests that the vte feature is not strongly related to age. This is because the age range for individuals with a VTE value of 1 and 0 is similar, which indicates that the VTE feature does not have a significant impact on age.

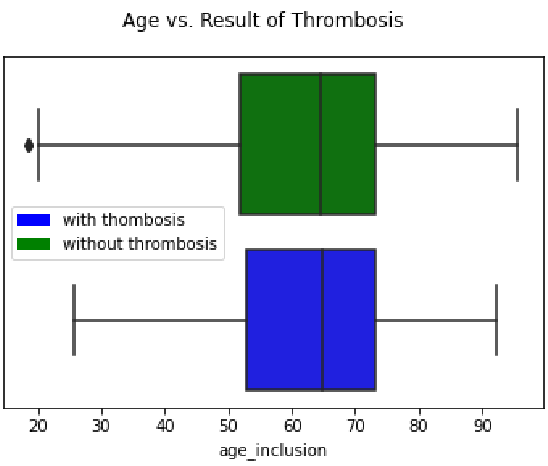


Figure 4.3: Box plot for the age vs. result of thrombosis

A relationship between the ”d-dimer” results and ”vte” is shown in Figure 4.3 as a multiple histogram, density estimate plot, and box plot to visualize the distribution of the d-dimer values for individuals with and without a vte. This plot shows that the d-dimer values for individuals with a vte value are concentrated in the range between 0.8 and 7.5, while the d-dimer values for individuals without a vte value are concentrated in the range below 0.8.

These findings suggest that the d-dimer values for individuals with a VTE value are higher than the d-dimer values for individuals without a VTE value. Additionally, the box plot shows that there are some individuals with a vte value who have d-dimer values in the range between 15 and 23. This could indicate that there is a subgroup of individuals with a high VTE value who have higher D-Dimer values than the other individuals in that category.

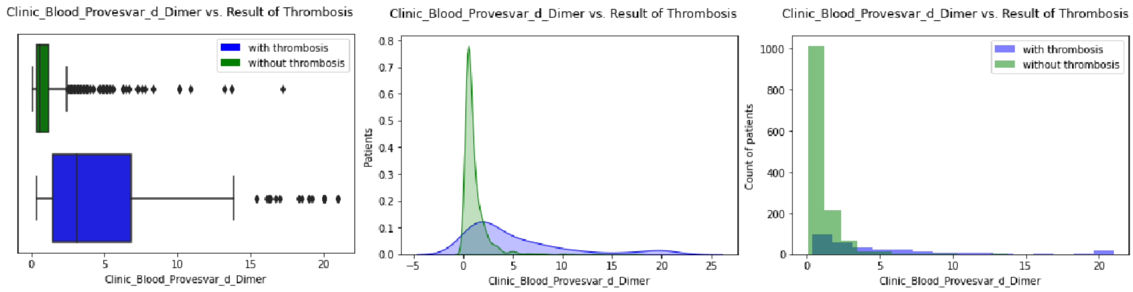


Figure 4.4: Visualizations for the D-Dimer vs. result of thrombosis

## 4.4 Machine Learning Setup

This research focuses on a classification problem. As we have 1 (thrombosis) or 0 (without thrombosis) as the labels, the techniques used for this research (mentioned in the Related Work chapter 2) will be compared, and the framework used will be consistent for both of the models. This section will elaborate on the things used in the experiments.

### 4.4.1 Data Preprocessing

Preparing a dataset for subsequent analysis involves a series of activities called data preprocessing. It can help to improve the quality of the data and make it more appropriate for the intended analysis, making it a crucial stage in the data analysis process.

The Ri-Schedule dataset needed several steps of preprocessing, including removing duplicates from the data, selecting the features, encoding the categorical features, imputing the missing values, and removing the outliers.

Data preprocessing is a crucial step in the analysis of any dataset because it can help improve the accuracy and reliability of the results. By cleaning, transforming, and scaling the data, we can ensure that the analysis is based on high-quality data that is free of errors or biases. Data preprocessing can help improve the validity of the results and provide more meaningful insights from the data.

**Removing Duplicates from the Ri-Schedule Dataset:** Duplicate records can occur for various reasons, such as errors in data entry or collection, multiple measurements of the same patient, or other issues.

These records can cause problems in machine learning because they can introduce bias and noise into the data and make it difficult to accurately evaluate the performance of the model. Therefore, it is a good idea to remove duplicate records from the dataset before using it for machine learning.

There are several ways to identify and remove duplicate records from a dataset. One common approach is to compute a unique identifier for each record, such as a patient ID in the record, and then use this identifier to identify and remove duplicates. Alternatively, we could compare the values in each record to identify duplicates based on specific criteria, such as matching values for certain key attributes [38].

Once the duplicate records have been identified, they were removed by dropping them from the dataset. The identifier used in the Ri-Schedule dataset to remove duplicates was the patient ID that was referring to the same entry [38].

**Selecting Features from the Ri-Schedule Dataset:** In order to build an effective and efficient machine learning model for predicting DVT using the Ri-Schedule dataset, it is important to carefully select the most relevant and informative features from the dataset.

Due to their lack of relevance to the study, relatively sparse data, follow-up data for patients with positively identified conditions, and contradicting data values, several of the variables from the original dataset were excluded. Left and right knee and ankle circumference measurement pairs were swapped out for new variables that contained the absolute values of the variance between the readings. An individual measurement of a limb does not currently have predictive value for the diagnosis of DVT, but a non-zero discrepancy between the two readings may indicate a swollen limb, which is one of the potential symptoms of a blood clot. This transformation was inspired by this fact. [38].

The dataset has 1653 samples and 159 independent variables after empty, sparse, or unnecessary features were removed. Personal characteristics, clinical symptoms, risk factors, vital signs, laboratory test results, limb measurements, follow-up visit information, prescribed medications, and diagnostic outcomes were among these variables. To choose the most pertinent and predicative characteristics for predicting DVT, a filter was used, and wrapper approaches were used for these variables[38].

**Converting Categorical Features from the Ri-Schedule Dataset:** One of the challenges in working with the Ri-Schedule dataset is that it contains both numerical and categorical data. The numerical data includes continuous and discrete values, such as age, weight, height, temperature, and laboratory test results. The categorical data includes nominal and ordinal values, such as gender, risk factors, vital signs, clinical symptoms, prescribed medications, and diagnostic results.

The categorical data must be encoded into a numerical format that the machine learning model can comprehend in order to be used in the model. Categorical data can be encoded in a variety of ways, including one-hot, binary, and label encoding. It's crucial to select the best strategy for the particular dataset and issue at hand because each of these has advantages and limitations [38].

In our case, the features were converted from categorical to dummy variables using one-hot encoding. By replacing the original variable with k-1 binary features, where k is the number of original categories, this transformation produces a binary representation of each predictor category. This method is suitable for nominal and ordinal data, and it allows the machine learning model to learn the inherent relationships between the different categories. It also avoids the issue of assigning arbitrary numeric values to the categories, which can introduce bias and reduce the model's performance [38].

After applying one-hot encoding to the categorical data in the Ri-Schedule dataset, we obtained a new dataset with expanded dimensions and 197 additional features. The use of one-hot encoding helps effectively incorporate the categorical data into the model and improve its performance.

**Imputing Missing Data in the Ri-Schedule Dataset:** Missing values occur when some of the attributes in a record are not available or are not recorded, and they can cause problems in the analysis and modeling of the data.

In order to address these challenges, several strategies were applied to handle missing values in the Ri-Schedule dataset. For missing values, imputation methods were used to fill in the gaps in the data. These methods included mean imputation, median imputation, and mode imputation, and the most appropriate method was selected for each attribute based on its characteristics and distribution. In general, where the predictor values were real numbers, the missing or incorrect values were replaced with means; the modes were imputed into binary-valued columns, and the medians were imputed into columns containing constrained sets of integer values [38].

After applying these strategies to the Ri-Schedule dataset, a clean and consistent dataset was obtained that was ready for further analysis and modeling. The handling of missing values was critical for building an accurate and reliable machine learning model for predicting DVT using the Ri-Schedule dataset. This process ensures that the data was complete, consistent, and free from errors, and it allows the machine learning model to learn from the most relevant and informative patterns in the data.

**Final Preprocessed Ri-Schedule Dataset:** After performing feature selection, handling missing values, and encoding categorical data, the final preprocessed dataset was obtained for the Ri-Schedule experiment. This dataset contained 1653 samples and 197 independent variables, and it represented a comprehensive and accurate representation of the data. The samples in the dataset were carefully selected and filtered to remove irrelevant, redundant, or conflicting information, and the remaining features were carefully cleaned and transformed to ensure their quality and consistency.

However, the final dataset had an unbalanced class distribution, with a negative-to-positive label ratio of 0.81:0.19. This is a common problem in machine learning, and it can affect the performance of the model if not properly addressed. To handle this issue, we adjusted the class weights, the weights for classes 0 and 1 were 0.61 and 2.65, respectively. This ensured that the model treated the two classes equally and learned from them in a balanced manner [38].

We randomly divided the dataset into training and test sets with an 80:20 ratio after modifying the class weights. We obtained training and test sets of 1322 and 331 samples, respectively, using stratified sampling to maintain the same distribution of classes in both sets. In the training and test subgroups, the negative-to-positive ratios were roughly constant at 0.81:0.19 [38].

In addition, we applied standardization to the dataset to transform the features into a common value range [0, 1]. This ensured that the features had similar scales and distributions, as it allows the model to learn from the data more effectively [38].

### 4.4.2 Performance Metrics:

Since we have a classification problem, we used accuracy, precision, recall, the F1 score, and area-under the curve-precision recall (AUC PR) for the evaluation of the model's performance in these investigations. These metrics are explained in the chapter 2

### 4.4.3 Hyperparameter Tuning

To find the best performing model Optuna was used to perform the hyperparameter serach on the training dataset, [19] we can have a dynamic search space with Optuna and use various pruning strategies. 5-fold stratified cross folds were used for the tuning, and the mean accuracy across the folds was used for the selection of the model.

Optuna is an open-source hyperparameter optimization framework designed to automate and optimize the process of machine learning model tuning. It provides an intuitive interface for defining optimization objectives and tuning parameters, as well as a powerful optimization engine for finding the optimal values of hyperparameters. With Optuna, we can quickly and easily optimize the performance of a machine learning model.

Optuna is based on a multi-armed bandit approach, which allows it to explore a large parameter space quickly and efficiently. This allows Optuna to effectively optimize a model's hyperparameters by testing a large number of different values and selecting the ones that yield the best results. Optuna also provides a variety of optimization algorithms, such as random search and Bayesian optimization, which helps users to find the optimal values of hyperparameters.

Optuna is also highly extensible and can be easily integrated with other machine learning frameworks such as TensorFlow, PyTorch, and Scikit-learn. This makes it easy to use Optuna to optimize their machine learning models in any environment. Additionally, Optuna has a number of built-in features that allows to quickly and easily debug and visualize their optimization process.

There are various other hyperparameter tuning techniques such as, bayesian optimization, grid search, and randomized search are all hyperparameter optimization techniques that are used to find the optimal values for a machine learning model's hyperparameters.

Bayesian optimization is a probabilistic model-based optimization technique that uses

Bayesian inference to update a posterior distribution of the function to be optimized. It assumes that the function to be optimized can be modeled as a sample from a Gaussian process. Bayesian optimization can be more efficient than other techniques because it uses prior knowledge about the function to guide the search for the optimal values of the hyperparameters.

As a brute-force approach to hyperparameter optimization, grid search consists of creating a grid of hyperparameter values, training models for each combination of those values, and then evaluating the performance of each model. Grid search is simple to implement and can be effective for low-dimensional problems with few hyperparameters, but it can be computationally expensive and may not be effective for larger, more complex problems.

A technique for hyperparameter optimization known as randomized search includes selecting random hyperparameter values from a given distribution, training a model, and assessing it for each set of sampled values. Because it doesn't involve training and testing a model for every possible combination of hyperparameter values, randomized search can be more effective than grid search, although it can still be computationally expensive for larger situations.

In comparison to these other techniques, Optuna offers several advantages. Optuna uses a Tree-structured Parzen Estimator (TPE) algorithm to guide the search for the optimal hyperparameters, which can be more efficient than other methods. Optuna also supports parallelization, which can speed up the optimization process. Additionally, Optuna has a user-friendly API and integrates with popular machine learning frameworks, making it easy to use and customize. Overall, Optuna can be a more efficient and effective method for hyperparameter optimization than other techniques such as Bayesian optimization, grid search, and randomized search.

Tree-structured Parzen estimator (TPE) is a probabilistic model-based optimization algorithm that is used in hyperparameter optimization. TPE is a variant of the Parzen-Rosenblatt estimator, which is a non-parametric density estimation algorithm. TPE uses Bayesian inference to model the distribution of the function to be optimized, and it uses a tree structure to guide the search for the optimal values of the hyperparameters.

The main advantage of using TPE for hyperparameter optimization is that it can be more efficient than other optimization algorithms. TPE uses prior knowledge about the function to guide the search for the optimal hyperparameters, which can help it avoid areas of the hyperparameter space that are unlikely to contain good solutions. This can make TPE faster and more effective than other optimization algorithms, especially for complex problems with many hyperparameters.

Additionally, TPE can be easily parallelized, which can further improve its efficiency. This makes TPE a good choice for optimizing machine learning models, where the optimization process can be computationally expensive. Overall, TPE is a powerful and effective algorithm for hyperparameter optimization that can help users find the optimal values for their machine learning models.

### 4.4.4 Feed-Forward Neural Network Parameters

There are several different parameters that can be tuned to control the behavior and performance of a neural network model. Some of these parameters include the following:

**The optimizer:** This is the algorithm used to optimize the weights of the network. TensorFlow provides several different optimizers, such as Adagrad, Adam, and RMSprop, each with its own set of hyperparameters that can be tuned. The optimal values for these hyperparameters depend on the specific task and dataset, but some recommended starting values include a learning rate of 0.01 for Adagrad, 0.001 for Adam, and 0.01 for RMSprop.

**The activation function:** This is the function used to compute the output of each neuron in the network. TensorFlow provides several different activation functions, such as ReLU, sigmoid, and tanh, each with its own characteristics and behavior.

**The number of layers and neurons:** The capacity and expressiveness of the network can be impacted by the number of layers and neurons in the network. Deeper networks with more layers and neurons can typically learn more complicated data patterns, but they can also be more computationally expensive and more prone to overfitting. 1-3 is a suggested starting point for the number of layers, and 10-100 is a suggested starting point for the number of neurons in each layer.

**The dropout:** This is a regularization technique for neural networks that helps prevent overfitting. It works by randomly dropping out, or setting to zero, a certain number of output units in a layer during training. This has the effect of "dropping out" or removing certain features or combinations of features, which helps to prevent overfitting. The recommended values to tune are between 0.1 and 0.7, which is the fraction of units that are dropped out in each layer. The optimal value will typically be determined through experimentation, using a validation set to evaluate the performance of the model with different dropout rates.

**The Learning Rate:** The learning rate regulates how frequently the optimizer updates the model's weights while it is being trained. Depending on this, the model will either learn from the data quickly or slowly. Although the training process will be slow with a low learning rate, the model will eventually converge to a more accurate result. On the other side, a high learning rate will speed up the training process but may lead to a less accurate solution or possibly a divergence in the model.

| Feed-Forward Neural Network | | | | | |
|---|---|---|---|---|---|
| Optimizer | Layers | Neurons | Activation Function | Dropout Rate | Learning Rate |
| Adam | 1 | 479 | sigmoid | 0.7 | 0.0321 |
| RMSprop | 2 | 161, 509 | relu | 0.1, 0.4 | 0.0421 |
| Adagrad | 3 | 493, 24, 329 | relu | 0.5, 0.1, 0.7 | 0.0551 |

Table 4.1: Tuned FFNN parameters from the hyperparameter search using Optuna.

### 4.4.5 TabNet Parameters

Similar to other machine learning models, TabNet also has some parameters that can be tuned, as elaborated in Chapter 2. The parameters tuned in our experimentation are:

**mask-type:** This is a masking function in the TabNet model to select features from the dataset. Masking can improve performance by excluding irrelevant or noisy input features, prevent overfitting by limiting the amount of information that the model can use, and increase interpretability by showing which input features the model is using to make predictions. There are two types of masking function that can be use, either ”entmax” or ”sparsemax”.

**n-da:** It is a decision prediction layer width. Increasing the width of the decision prediction layer can give the model more capacity to learn complex patterns, but it can also increase the risk of overfitting. Values for this hyperparameter typically range from 8 to 64.

**n-steps:** It is also a parameter in the model that refers to how many steps there are in the model's design. The number of steps is usually between 3 and 10. The number of steps in the architecture can affect the model's performance and accuracy; more steps may allow the model to recognize more intricate patterns in the data. However, using too many steps may make training the model more difficult and result in overfitting.

**gamma:** Gamma refers to the regularization parameter in the TabNet model. It is the coefficient for feature reusage in the masks. A value of gamma close to 1 would make mask selection least correlated between layers, which can help to prevent overfitting and improve the model's generalization ability. The range of possible values for gamma is typically from 1.0 to 2.0.

**n-shared:** The n-shared parameter in the TabNet model determines the number of shared ”Gated Linear Units” used to calculate the output of the network at each step. The n-shared parameter can be tuned to determine the overall complexity of the model. The recommended values range from 1 to 5.

**lambda-sparse:** The lambda sparse in the TabNet model is an extra sparsity loss coefficient that can be adjusted to influence the sparsity of the model in terms of feature selection. By increasing the coefficient, the model will select fewer features as it trains, thus reducing complexity. The value ranges from 1e-3 to 1e-6.

Default values were used for the rest of the parameters. The selected parameters for the tuned parameters are provided in table 4.2.

| TabNet | |
|---|---|
| Mask Type | sparsemax |
| n da | 64 |
| n steps | 6 |
| gamma | 1.4 |
| n shared | 2 |
| lambda sparse | 1.4658862255635718e-05 |
| patienceScheduler | 5 |
| patience | 20 |
| epochs | 86 |

Table 4.2: Tuned TabNet parameters from the hyperparameter search using Optuna.

# Chapter 5

# Results

This chapter presents the results obtained from the experimentation performed during the research.

## 5.1 Feed Forward Neural Networks Results

The results produced from the experimentation using FFNN are elaborated in this section.

The results obtained from different combinations of models based on varying optimizers during the hyperparameter search are presented in the table 5.1. The model that provided the best accuracy during the hyperparameter search was selected.

The best results are provided by the ”Adam” optimizer, with an accuracy of 84.89, followed by the optimizer ”RMSProp”, the accuracy was not significantly low but comparatively less than the best model; finally, the optimizer ”Adagrad” with the activation function ”relu” provided least accuracy among all the different variations.

| Feed-Forward Neural Network | |
|---|---|
| Optimizer | Accuracy |
| Adam | **85.49** |
| RMSprop | 84.89 |
| Adagrad | 80.96 |

Table 5.1: Result of optimzers for the FFNN models.

### 5.1.1 Confusion Matrix

The confusion matrix in Figure 5.1.1 summarizes the performance of a feed forward neural network model that attempted to predict whether a patient had thrombosis or not. In this matrix, the model is classifying patients as either having thrombosis or not having thrombosis.

The left column shows the actual condition of the patients (with or without thrombosis). The top row shows the model's prediction for each patient (whether it predicted the patient to have thrombosis or not). The cells contain the number of cases where the model correctly predicted the outcome (true positives and true negatives) and the number of cases where the model incorrectly predicted the outcome (false positives and false negatives).

In this matrix, the model correctly predicted that 257 patients did not have thrombosis (true negative), and that 26 patients did have thrombosis (true positive). However, the model incorrectly predicted that 12 patients did not have thrombosis (false negative) and 36 patients did have thrombosis (false positive).

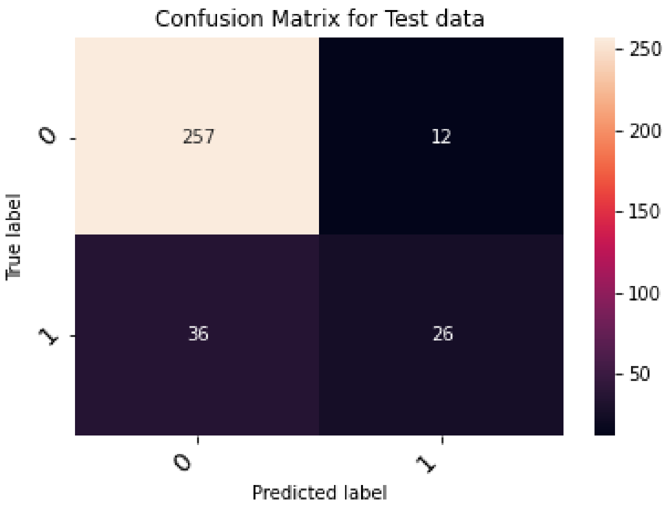


Figure 5.1: Confusion matrix indicating performance of FFNN on the test data for the classification of patients

### 5.1.2 Precision Recall Curves

The precision-recall curve for the FFNN model on the test set of the thrombosis dataset is presented in Figure 5.1.2. The line chart presents the combination of precision and recall for every threshold. In the beginning, it can be seen that the precision declined immediately, and after the best F1 score, it started to decline gradually. The best F1 score is a little bit to the left of the center of the chart, which means we have higher precision than the recall. The best threshold and F-Score were noted as 0.49 and 0.53, respectively.

The table 5.2 shows performance metrics calculated from the predictions of the final model that was selected from the hyperparameter search. From the results, it can be seen that there is a difference in the performance of the model on the training and test datasets; however, the overall results are relatively convincing since there is not any drastic difference in the performance of the model on both of these datasets.

## 5.2 TabNet Results

The results produced from the experimentation using TabNet are presented in this section

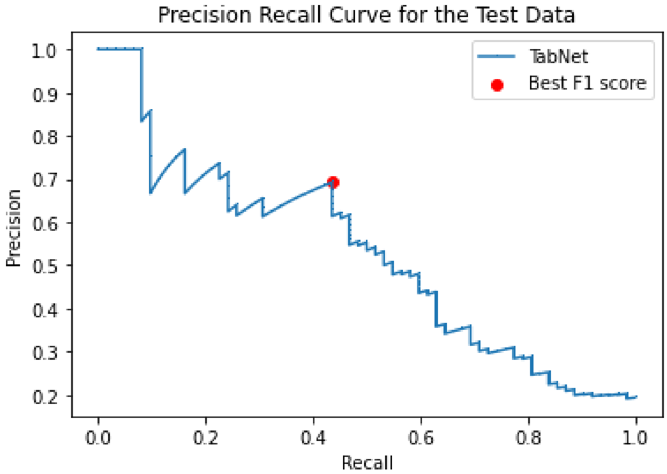


Figure 5.2: Precision-recall curve indicating performance of FFNN on the test data for the classification of patients

| Performance of the Final FFNN Model on Training and Test Set | | | | | |
|---|---|---|---|---|---|
| **Dataset** | **Accuracy** | **Precision** | **Recall** | **F1 Score** | **AUC (PR)** |
| Training | 88.95 | 0.83 | 0.51 | 0.63 | 0.92 |
| Test | 85.49 | 0.68 | 0.42 | 0.52 | 0.75 |

Table 5.2: Performance metrics from the prediction of thrombosis using the selected FFNN model.

### 5.2.1 Confusion Matrix

Figure 5.2.1 presents the confusion matrix for the TabNet model on the test set of the thrombosis dataset. The columns of the matrix represent the predicted class, and the rows represent the actual class. In this case, the two classes are ”without thrombosis” and ”thrombosis.”

The first row of the matrix shows the number of cases where the actual class was ”without thrombosis” and the predicted class was ”without thrombosis” (241 cases), and the number of cases where the actual class was ”without thrombosis” but the predicted class was ”thrombosis” (28 cases).

The second row shows the number of cases where the actual class was ”thrombosis” and the predicted class was ”without thrombosis” (28 cases), and the number of cases where the actual class was ”thrombosis” and the predicted class was ”thrombosis” (34 cases).

Overall, the confusion matrix shows that the model had a relatively high accuracy for predicting cases of thrombosis, but there were a significant number of false negatives (cases where the model predicted ”without thrombosis” but the actual class was ”thrombosis”).

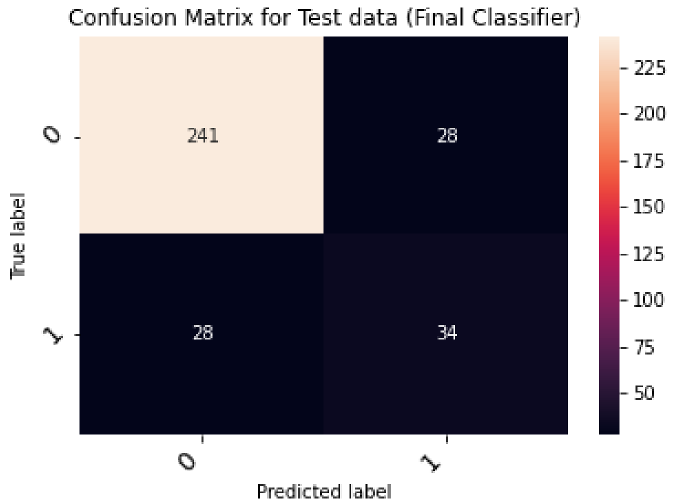


Figure 5.3: Confusion matrix indicating performance of TabNet on the test data for the classification of patients

### 5.2.2 Precision Recall Curves

Figure 5.2.2 shows the TabNet model's precision-recall curve for the test set of the thrombosis dataset. For each threshold, the line chart displays the sum of precision and recall. It can be seen that the precision initially reduced dramatically at one point before declining progressively. The model has done better as seen by the F1 score being to the right of the chart's center. The best threshold was determined to be 0.48, and the best F-Score was 0.57.

The table 5.3 lists performance indicators that were generated using the final model's

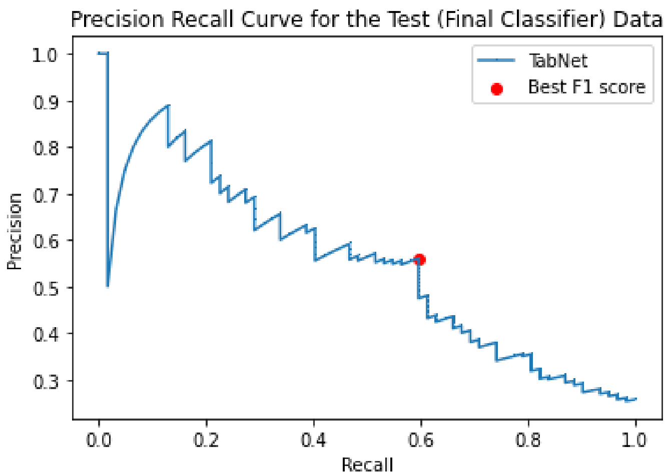


Figure 5.4: Precision-recall curve indicating performance of TabNet on the test data for the classification of patients

predictions, that was chosen through the hyperparameter search. The results show that the model performs differently on the training and test datasets; nonetheless, the overall findings are rather reasonable because there isn't a major difference in the model's performance on either of these datasets. Precision, recall, and F1 score are equal for the test set when the confusion matrix indicates that there are an equal amount of false positives and false negatives as shown in the confusion matrix 5.2.1.

| Performance of the Final TabNet Model on Training and Test Set | | | | | |
|---|---|---|---|---|---|
| **Dataset** | **Accuracy** | **Precision** | **Recall** | **F1 Score** | **AUC (PR)** |
| Training | 85.93 | 0.64 | 0.57 | 0.60 | 0.87 |
| Test | 83.08 | 0.54 | 0.54 | 0.54 | 0.82 |

Table 5.3: Performance metrics from the prediction of thrombosis using the selected TabNet model.

# Chapter 6

# Discussion

## 6.1 Research Question 1 - Performance of Feed-Forward Neural Network and TabNet

In this research, we used two different machine learning techniques, FFNN and TabNet. Both of these models were tuned so that maximum performance could be achieved from our dataset. The models were trained to efficiently predict patients with and without thrombosis.

Based on our evaluation using accuracy, recall, precision, and F1 score as metrics for the FFNN and TabNet models as shown in table 6.1, the results of the study indicate that the performance of the FFNN and TabNet models for the prediction of VTE was comparable, with the FFNN achieving an accuracy of 85.49 percent and the TabNet achieving an accuracy of 83.08 percent.

In terms of precision, the neural network outperformed the TabNet, achieving a precision

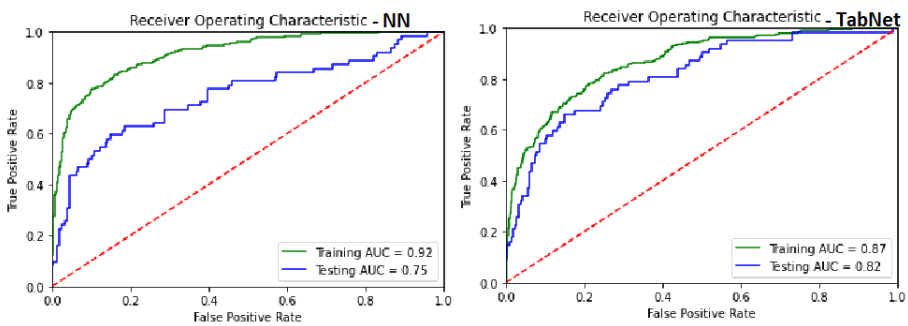


Figure 6.1: Receiver Operating Characteristic for FFNN and TabNet indicating performance on the training and test data for the classification of thrombosis

score of 0.68 compared to 0.54 for the TabNet. Precision measures the proportion of true positive predictions made by the model, so a higher precision score indicates that the model is more accurate in predicting positive cases.

In terms of recall, the TabNet performed better as well, achieving a recall score of 0.54

compared to 0.42 for the neural network. Recall measures the proportion of positive cases that were correctly predicted by the model, so a higher recall score indicates that the model is more effective at identifying positive cases.

The F1 score, which is the harmonic mean of precision and recall, was also similar for the two models, with the neural network achieving an F1 score of 0.52 and the TabNet achieving an F1 score of 0.54.

Finally, in terms of the AUC (PR) metric, which measures the model's ability to distinguish between positive and negative cases, the neural network performed slightly worse than the TabNet, achieving an AUC (PR) score of 0.75 compared to 0.82 for the TabNet.

The results of the study suggest that the performances of the FFNN and TabNet models for the classification of patients with or without VTE were quite similar, with the FFNN achieving slightly better results in terms of precision and the TabNet achieving slightly better results in terms of AUC (PR).

| Performance of the Final NN Model on Training and Test Set | | | | | |
|---|---|---|---|---|---|
| **Dataset** | **Accuracy** | **Precision** | **Recall** | **F1 Score** | **AUC (PR)** |
| Training | 88.95 | 0.83 | 0.51 | 0.63 | 0.92 |
| Test | 85.49 | 0.68 | 0.42 | 0.52 | 0.75 |
| Performance of the Final TabNet Model on Training and Test Set | | | | | |
| **Dataset** | **Accuracy** | **Precision** | **Recall** | **F1 Score** | **AUC (PR)** |
| Training | 85.93 | 0.64 | 0.57 | 0.60 | 0.87 |
| Test | 83.08 | 0.54 | 0.54 | 0.54 | 0.82 |

Table 6.1: Performance of FFNN and Tabnet from the prediction of thrombosis.

## 6.2 Research Question 2 - Performance of FFNN and TabNet in Minimizing False Negatives

The figure 6.2 shows the precision-recall curves for both the FFNN and TabNet models when we slide to threshold to have the best recall at a fixed threshold of 1.0, we can see the tradeoff between precision and the recall, for the FFNN model, we got precision of 0.20 whereas TabNet managed to get 0.15 of the precision.

When we have the fixed threshold at 1.0 for the recall this implies both TabNet and FFNN achieved perfect recall, successfully identifying every positive event in the test set. The precision at this level varies across the two models, though. Only 20% of the positive predictions made by FFNN are accurate, according to its precision score of 0.20. When compared to TabNet, which has a precision of 0.15, only 15% of the positive predictions are accurate. According to these findings, FFNN outperforms TabNet in terms of precision at this particular recall threshold.

Based on the confusion matrices in Figures 5.1.1 and 5.2.1, the performances of both the FFNN and TabNet in minimizing false negatives can be compared. In the case of

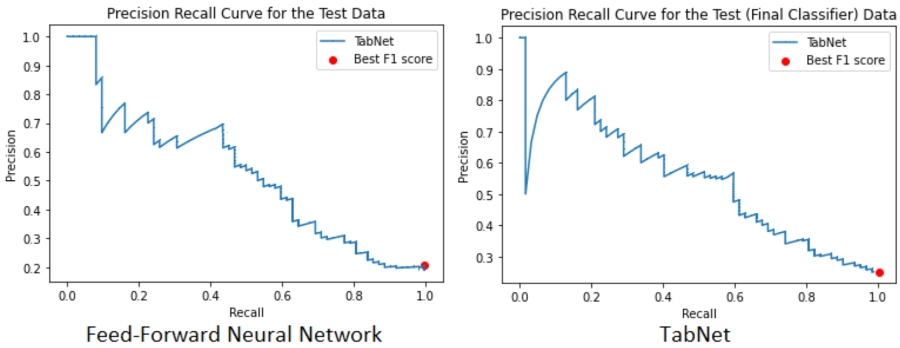


Figure 6.2: Precision-recall curves for FFNN and TabNet with best recall

the FFNN, there were 12 false negatives, meaning that there were 12 instances in which the model predicted that a patient did not have thrombosis when they actually did. In the case of TabNet, there were 28 false negatives, meaning that there were 28 instances in which the model predicted that a patient did not have thrombosis when they actually did.

When considering the overall performance of the two models in minimizing false negatives, it appears that the FFNN was more successful at this task. The FFNN had only 12 false negatives, while TabNet had 28 false negatives. This suggests that the FFNN was more effective at minimizing false negatives than TabNet, as it had fewer false negatives.

However, it is important to note that both models had a fairly low rate of false negatives, which is generally a good sign of performance.

# Chapter 7

# Conclusion

This chapter presents the summarized main findings and implications of the research. In this chapter, we will review the results of our comparison between the FFNN and the TabNet, and discuss the implications of these results for future research.

## 7.1 Conclusions

We compared the performance of two different machine learning models: a FFNN and a TabNet in this study. We evaluated their performance using a variety of metrics, including accuracy, precision, recall, F1 score, and AUC (PR).

The results of our analysis showed that the FFNN outperformed the TabNet in most metrics. FFNN achieved an accuracy of 85.49 percent, a precision of 0.68, a recall of 0.42, and an F1 score of 0.52. The TabNet, on the other hand, achieved an accuracy of 83.08 percent, a precision of 0.54, a recall of 0.54, and an F1 score of 0.54. Additionally, the AUC (PR) for the neural network was 0.75, while the AUC (PR) for the TabNet was 0.82.

We also examined the confusion matrices for each model to determine their performance at minimizing false negatives. The confusion matrix for the FFNN showed that it had a lower number of false negatives (12) compared to the TabNet (28). This indicates that the neural network is better at predicting the positive class.

The results of this study suggest that the FFNN is the better performing model in terms of the metrics we evaluated.

## 7.2 Future Work

A FFNN and a TabNet were the two machine learning models that we compared in this study for performance. While the results of our examination showed that FFNN outperformed TabNet by a slight margin, more analysis is required to completely comprehend the strengths and weaknesses of both models.

Expanding the dataset utilized in this study and evaluating the models on a wider and more varied set of data is one potential area for future research. This might offer a more thorough evaluation of the models' performance and assist in identifying any potential biases or shortcomings.

The use of aggressive hyperparameter settings for the FFNN and the TabNet using GridSearch is another potential route for future research, as this brute-force strategy may yield a model that is more suited for this application. This might improve the capabilities of these models and make it possible for them to handle the dataset's more difficult problems.

# Bibliography


[1] W. Spangler, "The role of artificial intelligence in understanding the strategic decision-making process," *IEEE Transactions on Knowledge and Data Engineering*, vol. 3, no. 2, pp. 149–159, 1991. DOI: 10.1109/69.87995.

[2] N. J. Nilsson, "Introduction to machine learning. an early draft of a proposed textbook," 1996.

[3] G. F. Cooper, C. F. Aliferis, R. Ambrosino, *et al.*, "An evaluation of machine-learning methods for predicting pneumonia mortality," *Artificial intelligence in medicine*, vol. 9, no. 2, pp. 107–138, 1997.

[4] F. R. Rosendaal, "Venous thrombosis: A multicausal disease," *The Lancet*, vol. 353, no. 9159, pp. 1167–1173, 1999.

[5] J. Hirsh and A. Y. Lee, "How we diagnose and treat deep vein thrombosis," *Blood, The Journal of the American Society of Hematology*, vol. 99, no. 9, pp. 3102–3110, 2002.

[6] S. P. Jackson, "Arterial thrombosis—insidious, unpredictable and deadly," *Nature medicine*, vol. 17, no. 11, pp. 1423–1436, 2011.

[7] M. Levi, T. van der Poll, and M. Schultz, "Infection and inflammation as risk factors for thrombosis and atherosclerosis," in *Seminars in thrombosis and hemostasis*, Thieme Medical Publishers, vol. 38, 2012, pp. 506–514.

[8] W. N. Venables and B. D. Ripley, *Modern applied statistics with S-PLUS*. Springer Science & Business Media, 2013.

[9] You and B. Clots. "Why is timely diagnosis of DVT and PE important." (2016), [Online]. Available: https://www.youandbloodclots.com/en-vte/view/m101-e07-why-is-timely-diagnosis-of-dvt-and-pe-important-expert-video#:~:text=Prompt%5C%20diagnosis%5C%20and%5C%20treatment%5C%20can,early%5C%20is%5C%20your%5C%20best%5C%20defense (visited on 03/16/2022).

[10] B. J. Erickson, P. Korfiatis, Z. Akkus, and T. L. Kline, "Machine learning for medical imaging," *Radiographics*, vol. 37, no. 2, p. 505, 2017.

[11] Y. Fei, J. Hu, K. Gao, J. Tu, W.-q. Li, and W. Wang, "Predicting risk for portal vein thrombosis in acute pancreatitis patients: A comparison of radical basis function artificial neural network and logistic regression models," *Journal of critical care*, vol. 39, pp. 115–123, 2017.

[12] F. Jiang, Y. Jiang, H. Zhi, *et al.*, "Artificial intelligence in healthcare: Past, present and future," *Stroke and vascular neurology*, vol. 2, no. 4, 2017.

[13] S. Nayak and D. Gope, "Comparison of supervised learning algorithms for rf-based breast cancer detection," in *2017 Computing and Electromagnetics International Workshop (CEM)*, IEEE, 2017, pp. 13–14.

[14] M. Nilashi, O. bin Ibrahim, H. Ahmadi, and L. Shahmoradi, "An analytical method for diseases prediction using machine learning techniques," *Computers & Chemical Engineering*, vol. 106, pp. 212–223, 2017.

[15] F. Osisanwo, J. Akinsola, O. Awodele, J. Hinmikaiye, O. Olakanmi, and J. Akinjobi, "Supervised machine learning algorithms: Classification and comparison," *International Journal of Computer Trends and Technology (IJCTT)*, vol. 48, no. 3, pp. 128–138, 2017.

[16] R. Saravanan and P. Sujatha, "A state of art techniques on machine learning algorithms: A perspective of supervised learning approaches in data classification," in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, 2018, pp. 945–949.

[17] K.-H. Yu, A. L. Beam, and I. S. Kohane, "Artificial intelligence in healthcare," *Nature biomedical engineering*, vol. 2, no. 10, pp. 719–731, 2018.

[18] W. Yue, Z. Wang, H. Chen, A. Payne, and X. Liu, "Machine learning with applications in breast cancer diagnosis and prognosis," *Designs*, vol. 2, no. 2, p. 13, 2018.

[19] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019.

[20] C. Clinic. "Hemostasis." (2019), [Online]. Available: `https://my.clevelandclinic.org/health/symptoms/21999-hemostasis#:~:text=Hemostasis%5C%20is%5C%20your%5C%20body's%5C%20natural,much%5C%20or%5C%20too%5C%20little%5C%20clotting` (visited on 03/16/2022).

[21] C. for Disease Control and Prevention. "Data and Statistics on Venous Thromboembolism." (2019), [Online]. Available: `https://www.cdc.gov/ncbddd/dvt/data.html#:~:text=The%5C%20precise%5C%20number%5C%20of%5C%20people,within%5C%20one%5C%20month%5C%20of%5C%20diagnosis` (visited on 03/16/2022).

[22] S. Liu, F. Zhang, L. Xie, *et al.*, "Machine learning approaches for risk assessment of peripherally inserted central catheter-related vein thrombosis in hospitalized patients with cancer," *International Journal of Medical Informatics*, vol. 129, pp. 175–183, 2019.

[23] J. H. Medicine. "Thrombosis." (2019), [Online]. Available: `https://www.hopkinsmedicine.org/health/conditions-and-diseases/thrombosis#:~:text=Thrombosis%5C%20occurs%5C%20when%5C%20blood%5C%20clots%5C%20block%5C%20veins%5C%20or%5C%20arteries.,a%5C%20stroke%5C%20or%5C%20heart%5C%20attack` (visited on 03/16/2022).

[24] N. R. Moșteanu, "International financial markets face to face with artificial intelligence and digital era.," *Theoretical & Applied Economics*, vol. 26, no. 3, 2019.

[25] J. I. Weitz and N. C. Chan, "Advances in antithrombotic therapy," *Arteriosclerosis, Thrombosis, and Vascular Biology*, vol. 39, no. 1, pp. 7–12, 2019.

[26] S. Doi, Y. J. Akashi, M. Takita, *et al.*, "Preventing thrombosis in a covid-19 patient by combined therapy with nafamostat and heparin during extracorporeal membrane oxygenation," *Acute medicine & surgery*, vol. 7, no. 1, e585, 2020.

[27] A. Lichota, E. M. Szewczyk, and K. Gwozdzinski, "Factors affecting the formation and treatment of thrombosis by natural and synthetic compounds," *International journal of molecular sciences*, vol. 21, no. 21, p. 7975, 2020.

[28] Y. Ma, Z. Wang, H. Yang, and L. Yang, "Artificial intelligence applications in the development of autonomous vehicles: A survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 2, pp. 315–329, 2020. DOI: 10.1109/JAS.2020.1003021.

[29] T. D. Martins, J. M. Annichino-Bizzacchi, A. V. C. Romano, and R. Maciel Filho, "Artificial neural networks for prediction of recurrent venous thromboembolism," *International journal of medical informatics*, vol. 141, p. 104 221, 2020.

[30] T. Nafee, C. M. Gibson, R. Travis, *et al.*, "Machine learning to predict venous thrombosis in acutely ill medical patients," *Research and practice in thrombosis and haemostasis*, vol. 4, no. 2, pp. 230–237, 2020.

[31] K. Abbas. "Predicting thrombosis with machine learning." (2021), [Online]. Available: https://hiof.brage.unit.no/hiof-xmlui/handle/11250/2770341 (visited on 02/12/2023).

[32] S. Ö. Arik and T. Pfister, "Tabnet: Attentive interpretable tabular learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 6679–6687.

[33] H. H. Rashidi, N. Tran, S. Albahra, and L. T. Dang, "Machine learning in health care and laboratory medicine: General overview of supervised learning and auto-ml," *International Journal of Laboratory Hematology*, vol. 43, pp. 15–22, 2021.

[34] L. Ryan, S. Mataraso, A. Siefkas, *et al.*, "A machine learning approach to predict deep venous thrombosis among hospitalized patients," *Clinical and Applied Thrombosis/Hemostasis*, vol. 27, p. 1 076 029 621 991 185, 2021.

[35] K. Zifkos, C. Dubois, and K. Schäfer, "Extracellular vesicles and thrombosis: Update on the clinical and experimental evidence," *International Journal of Molecular Sciences*, vol. 22, no. 17, p. 9317, 2021.

[36] V. Danilatou, S. Nikolakakis, D. Antonakaki, *et al.*, "Outcome prediction in critically-ill patients with venous thromboembolism and/or cancer using machine learning algorithms: External validation and comparison with scoring systems," *International Journal of Molecular Sciences*, vol. 23, no. 13, p. 7132, 2022.

[37] C. Shah, Q. Du, and Y. Xu, "Enhanced tabnet: Attentive interpretable tabular learning for hyperspectral image classification," *Remote Sensing*, vol. 14, no. 3, p. 716, 2022.

[38] R. Sorano, L. V. Magnusson, and K. Abbas, "Comparing effectiveness of machine learning methods for diagnosis of deep vein thrombosis," in *International Conference on Computational Science and Its Applications*, Springer, 2022, pp. 279–293.

[39] Y. E. Sun, H. K. Na, S. Kwak, Y. D. Kim, H. S. Nam, and J. H. Heo, "Different thrombus histology in a cancer patient with deep vein thrombosis and recurrent strokes," *Journal of Stroke*, vol. 24, no. 2, pp. 300–302, 2022.