Introduction

Venous thrombus is a cardiovascular disease that affects both children and adults. Deep Venous Thrombus (DVT) and Pulmonary Embolism (PE) are the two most common types of VTE. Many people die due to VTE globally. It is the number one cause of sudden deaths in patients. Nearly 300,000 people die due to PE in USA (Raskob, et al., 2014) and around 370,000 deaths were reported in due to VTE in 6 European countries in a single year (Farge, et al., 2019). A major problem with VTE is that its diagnostic accuracy is very low and VTE goes undetected in many patients, which proves fatal. Timely diagnose can help us save a lot of lives. Timely detection will allow proper treatment of patients & will reduce risks of death.

Several paragraphs have been removed.

7 Research

7.1 Material and Methods

7.1.1 Literature Search

We used PubMed to search literature on Artificial Intelligence on diagnostic of venous thrombosis. We retrieved literature from January 1st, 2000, till October 14th, 2021. We used “((("embolism and thrombosis"[MeSH Terms])) AND "medical records"[MeSH Terms]) AND "computing methodologies"[MeSH Terms]” to search all the relevant publications.

7.1.2 Study Selection

The article inclusion criteria were as per the following: (1) the study explored the utilization of AI/ML to diagnose venous thrombosis; (2) the study made use of patient journals to develop the algorithm.

The avoidance measures were as per the following: (1) purely theoretic studies; (2) studies without usable results; (3) nonhuman studies.

7.1.3 Data Extraction

All the relevant information such as type of venous thrombosis, AI methods used, clinical information, dataset used, and results were extracted from studies.

7.2 Results

A table has been removed.

We initially retrieved 52 studies from PubMed. After careful assessment according to needs of our study, we shortlisted 10 studies for assessment. We assessed studies based on US, Canada, Europe, China. It covered major parts of the world.

All the studies used both the training and testing dataset. It is observed in many studies that they collected many health records, and a subset of these records was used by authors. Half of selected studies used either 5-fold cross validation or 10-fold cross validation.

In the article “Ontology-based venous thromboembolism risk assessment model developing from medical records**”** the researchers used Machine Learning and ontology-based models to detect thromboembolism risk in patients. He used 3106 medical records from Peking Union Medical College Hospital (PUMCH). It contained admission note and progress note for each patient, and both were unstructured & contained free hand text. After pre-processing he used 156,918 unique terms for his ontology set. They used word2vec and continuous Skip gram model to obtain vector representation of words. They employed Random Forest (RF), Gradient Boosting decision trees (GBDT), Linear Regression (LR), and SVM and reported mean and standard deviation of AUC scores. Gradient boosted decision tree (GBDT) based model was best performing model. They also stated that ontology-based feature enrichment is superior to TF-IDF and removing noises are very important(Yang, et al., 2019).

Several paragraphs have been removed.

(Kawaler, et al., 2012) used two representations of data set. One unabridged data and a curated dataset of 119 variables depicting risk factors associated with VTE. They used many algorithms like Naïve Bayes, K nearest neighbours, SVM, C 4.5 and Random Forests. They also used ensemble techniques i.e., bagging and boosting of several best performing models. It did not affect the performance of these models significantly, so the authors omitted ensemble results. SVM, Naïve Bayes and Random Forest were the best performing algorithms. They were able to find high risk patients with high accuracy. Their models did well without having any prior knowledge about VTE & exceeded accuracy with previous scoring systems.

7.3 Research conclusion

We conducted literature review to analyse the current state of Artificial Intelligence for thrombosis detection. SVM is most widely used ML algorithm for PE and VTE detection. Multiple studies suggested use of NLP to enhance prediction capabilities of ML and DL algorithms. In our study many possible combinations of NLP, ML & DL were evaluated. NLP enriched EHR can be used to predict PE and DVT with good accuracy. Many methodologies were comparable with human classification levels. Deep Learning based CT image models generalised well and were enhanced using NLP.

NLP enrichment also provided some deterministic factors like diabetes, obesity, lack of sleep etc. which caused thrombosis. It is also helpful for doctors & patients as controlling these factors will reduce risks of disease like thrombosis. All the literature we studied provided some successful approaches to predict thrombosis. It is clear that ML and DL can be used to effectively detect thrombosis. We will use NLP methods along with ML and DL algorithms to make our own algorithm for detecting thrombosis. From the study it is clear that SVM and DL methods will be best to experiment with. We will use k-fold cross-validation to improve our model performance. There is still a lot of scope for research that could be worked on onwards.

8 Implementation

8.1 Data formatting and pre-processing

We performed our experiments using our datasets of patient EHR. The dataset was in the Danish language, as such, there were limitations in the methods we could use as most tools provided support for only the English language.

In our experiments, we focused on two columns in the dataset. The first had the details of the patient record. The second column contained the label for each record and was either “negative” or “positive” depending on whether VTE was detected or not. Each algorithm was then trained to take the text from the first column as input and then predict the corresponding label for that record as output.

Several paragraphs have been removed.

8.2 Algorithms

We trained five machine learning algorithms using Python Scikit-Learn library implementation and two deep learning algorithms with Python Keras Framework, Tensorflow backend. We used their default parameters except stated otherwise.

8.2.1 Naïve Bayes

Several paragraphs have been removed.

8.2.2 Support Vector Machine (SVM)

This algorithm is trained to find an optimal decision boundary (called a hyper-plane) that best separates the training data into two classes, such that the distance between the training data points to the hyper-plane (called the margin) is maximized (Vapnik, 1999). We trained our model using a linear SVM which assumes the training data to be linearly separable. We used a regularization parameter (called C) of 1.0, which controls the trade-off between a smooth decision boundary and classifying the training points correctly. Large values of C will choose a hyper-plane with smaller margins, while small values of C will choose a hyper-plane with larger margins even at the expense of misclassifying more data points.

8.2.3 Decision Tree

Several paragraphs have been removed.

8.2.4 Random Forest

Several paragraphs have been removed.

8.2.5 K Nearest Neighbour

Several paragraphs have been removed.

8.2.6 CNN

Several paragraphs have been removed.

8.2.7 LSTM

Several paragraphs have been removed.

8.3 Results

After training, we evaluated the performance of the different classifiers using our testing dataset. The SVM classifier achieved the best performance with an accuracy of 92.56%. The result is in the table below.

A table has been removed.

The SVM algorithm achieved superior performance over the other algorithms with an accuracy of 92.56%. We attribute this to the fact that most text classification problems are linearly separable, which is one of the assumptions of a linear SVM algorithm. (Joachims, 2001) noted also that SVM tends to be better at handling data with high dimensionality and sparseness.

Several paragraphs have been removed.

The K Nearest Neighbors algorithm performed the worst with an accuracy of 69.32%. A not surprising result, as the algorithm is sensitive to irrelevant features and generally does not work well with high-dimensional data (Pestov, 2013). Additional pre-processing steps, like stop-word removal and lemmatization, could be employed to reduce these irrelevant words and the feature size of the data.

Several images and paragraphs have been removed.

8.4 Conclusion

In this section, we discussed the pre-processing steps, algorithms, and the results from our experiments. We showed that even with minimal hyper-parameter tuning and limited support for the Danish language, we were able to achieve good performance from training machine learning algorithms on only a patients’ EHR in predicting whether they have VTE or not.

In addition, we described each algorithm in detail, compared our expectations to the actual results, and mentioned ways to improve them. We see it as very important for medical researchers without a previous machine learning experience.

We strongly believe that better performance can be obtained by spending some time tuning each algorithm's hyper-parameters and exploring other preprocessing methods.