DOKUZ EYLÜL UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

AN EXPLAINABLE AI APPLICATION FOR CLASSIFICATION OF BIOMEDICAL TEXT WITH A HYBRID APPROACH USING WORD EMBEDDING AND BAG-OF-WORD

by

Nizar Abdulaziz Mahyoub AHMED

October, 2022

İZMİR

AN EXPLAINABLE AI APPLICATION FOR CLASSIFICATION OF BIOMEDICAL TEXT WITH A HYBRID APPROACH USING WORD EMBEDDING AND BAG-OF-WORD

A Thesis Submitted to the
Graduate School of Natural And Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Computer Engineering Department

by

Nizar Abdulaziz Mahyoub AHMED

October, 2022
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled "**AN EXPLAINABLE AI APPLICATION FOR CLASSIFICATION OF BIOMEDICAL TEXT WITH A HYBRID APPROACH USING WORD EMBEDDING AND BAG-OF-WORD**" completed by **NIZAR ABDULAZIZ MAHYOUB AHMED** under supervision of **PROF. DR. ADİL ALPKOÇAK** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

.....

Prof. Dr. Adil ALPKOÇAK

Supervisor

.....	.....
Assoc. Prof. Dr. Zerrin IŞIK	Prof. Dr. Mehmet KUNTALP
Thesis Committee Member	Thesis Committee Member
.....	.....
Assoc. Prof. Dr. Belgin ERGENÇ	Assoc. Prof. Dr. Deniz KILINÇ
Examining Committee Member	Examining Committee Member

Prof.Dr. Okan FISTIKOĞLU
Director
Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

First and foremost, praises and thanks to the God, the Almighty, for his blessings throughout my research work to complete the research successfully.

I would like to express my deep and sincere gratitude to my supervisor, Prof. Dr. Adil ALPKOÇAK for giving me the opportunity to do this research and providing invaluable guidance throughout this research. His vision, sincerity, creativity and motivation have deeply inspired me. He has taught me each step during this great journey. More importantly, I'm so horned that together we have published several papers in some interested AI, Machine Learning and Deep learning topics. It was a great privilege and honor to work and study under his guidance. I would also like to thank him for his friendship, empathy, and great sense of humor.

I also want to dedicate my work to my deceased parents, who were the two most significant individuals in my life. They were the reason of what I became. Their prayers and support have been with me throughout my life. Additionally, I want to dedicate this work to my family and close friends who also have been a great support in my life.

I am extending my thanks to management of Graduate School of Arts and Sciences for their support during my research work. I also thank all the staff of Research section of Dokuz Eylül university for their kindness.

Finally, my thanks go to YTB, Yurtdışı Türkler ve Akraba Topluluklar Başkanlığı, for their emotional and financial support. I'm so grateful of granting this scholarship and giving me the opportunity of fulfilling one of the most important accomplishments in my life.

Nizar Abdulaziz Mahyoub AHMED

AN EXPLAINABLE AI APPLICATION FOR CLASSIFICATION OF BIOMEDICAL TEXT WITH A HYBRID APPROACH USING WORD EMBEDDING AND BAG-OF-WORD

ABSTRACT

Since most of explainable AI system (XAI) evaluation measures in literature need human intervention and are subjective, the need for unbiased and more automatic methods is indispensable. This thesis proposes an XAI accompanied by a multi-class classification model using a hybrid feature representation method. We developed three quantitative XAI evaluation measures that objectively calibrate the complexity of two XAI methods, which are SHAP and LIME.

In that context, we create a hybrid feature representation method that is used as a prepossessing step in a deep learning multi-classification system, combining both bag of words (BoW) and word embedding (WE) techniques. To this end, we use a bidirectional long-short-term memory (LSTM) deep learning model for the system. Thereafter, we implement several XAI methods, i.e. local and global XAI, on the outputs of the multi-classification system in order to explain the decisions of the employed model. Furthermore, we apply our proposed model and designed system to a medical multi-class benchmark called OHSUMED. Finally, we compare the complexity of the explainability between SHAP and LIME methods using the depth of the decision tree as an objective XAI evaluation measure.

The results of the new quantitative and automatic measures proposed by us show that SHAP outperforms LIME method because it is less complex. Additionally, based on SHAP feature importance scores, we become able to identify which features are the most significant and hence must be included in the multi-class classification model. As a result, using both the hybrid feature representation system as a prepossessing step and our proposed XAI evaluation measures, the accuracy of the classification model increased from 45.4% to 92%.

Keywords: Explainable AI, medical multi-class classification, quantitative explainability evaluation, word embedding, bag of words, deep learning,

BİYOMEDİKAL METİNLERİN SÖZCÜK İLİŞTİRME VE SÖZCÜK TORBASI YÖNTEMİ KULLANARAK MELEZ YAKLAŞIMLA SINIFLANDIRILMASINDA AÇIKLANABİLİR BİR YAPAY ZEKA UYGULAMASI

ÖZ

Literatürdeki açıklanabilir AI sistemi (XAI) değerlendirme önlemlerinin çoğu insan müdahalesine ihtiyaç duyduğundan ve öznel olduğundan, tarafsız ve daha otomatik yöntemlere duyulan ihtiyaç vazgeçilmezdir. Bu tez, hibrit bir özellik temsil yöntemi kullanan çok sınıflı bir sınıflandırma modelinin eşlik ettiği bir XAI önermektedir. SHAP ve LIME olan iki XAI yönteminin karmaşıklığını objektif olarak kalibre eden üç nicel XAI değerlendirme ölçütü geliştirdik.

Bu bağlamda, hem kelime çantası (BoW) hem de kelime gömme (WE) tekniklerini birleştiren, derin öğrenme çoklu sınıflandırma sisteminde önceden belirleme adımı olarak kullanılan hibrit bir özellik temsil yöntemi oluşturuyoruz. Bu amaçla, sistem için çift yönlü uzun kısa süreli bellek (LSTM) derin öğrenme modeli kullanıyoruz. Daha sonra, kullanılan modelin kararlarını açıklamak için çoklu sınıflandırma sisteminin çıktıları üzerinde yerel ve global XAI gibi çeşitli XAI yöntemlerini uyguladık. Ayrıca, önerilen modelimizi ve tasarlanan sistemimizi OHSUMED adlı tıbbi çok sınıflı bir kıyaslamayı uyguluyoruz. Son olarak, karar ağacının derinliğini objektif bir XAI değerlendirme ölçüsü olarak kullanarak SHAP ve LIME yöntemleri arasındaki açıklanabilirliğin karmaşıklığını karşılaştırıyoruz.

Tarafımızdan önerilen yeni nicel ve otomatik önlemlerin sonuçları, SHAP'ın daha az karmaşık olduğu için LIME yönteminden daha iyi performans sağladığını göstermektedir. Ek olarak, SHAP özellik önem puanlarına dayanarak, hangi özelliklerin en önemli olduğunu ve dolayısıyla çok sınıflı sınıflandırma modeline dahil edilmesi gerektiğini belirleyebiliriz. Sonuç olarak, hem hibrit öznitelik temsil sistemini bir ön hazırlık aşaması olarak hem de bizim önerilen XAI değerlendirme önlemlerimizi kullanarak, sınıflandırma modelinin doğruluğu %45,4'ten %92'ye

yükseldi.

Anahtar kelimeler: Açıklanabilir AI sistemi, tıbbi çok sınıflı sınıflandırma, objektif açıklanabilirlik değerlendirmesi, kelime gömme, kelime çantası, derin öğrenmesi.

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER ONE
INTRODUCTION

1.1 Overview

Text classification (TC) has evolved over the last decade to become one of the most fascinating areas of machine learning. The most engaging challenge in TC is assigning classes to unseen data by discovering patterns that each class can share. However, before categorizing the data, a classification model must first go through a stage known as feature representation. Feature representation is the process of converting text data into a form that the classifier can recognize, such as floating-point numerical vectors (Kowsari et al., 2019). The better the classifier's ability to discover patterns in the data, the more efficient the feature representation method used (Kowsari et al., 2019) (Sinoara et al., 2019).

The two most common feature representation techniques in TC are bag of words (BoW) and word embedding (WE). These two techniques are effective in classification systems, but their working mechanisms differ. Bag of words (BoW) is a method that represents the entire body text as a list of words, whether it is a document or a sentence. These words are saved in a matrix to be calculated later, regardless of their sentence-like form or syntax, grammar, or semantic relationships between them. Different weighting schemes for BoW are term frequency (TF) and term frequency-inverse document frequency (TF-IDF). However, word embedding (WE) techniques can discover syntax and context relationships between words (Kowsari et al., 2019) (Sinoara et al., 2019). Combining both techniques has significant advantages in representing data features, but combining them may yield a more powerful weighted feature representation model. As a result, when developing the classification system, the benefits of both techniques can be considered. As a result, many classification systems, such as multi-class and multi-label classification systems, can use advanced weighted feature representation.

Some people may mix up multi-class and multi-label systems, believing they are

the same thing. However, these systems are diametrically opposed. A multi-class classification system attempts to classify documents into a single class from multiple (more than two) classes (Lei et al., 2019). whereas multi-label systems assign one or more classes to a specific document (Blanco et al., 2020). Class imbalance, dealing with a large number of classes, and label dependency are all issues that both classification systems face. As a result, both classification systems attempt to highlight those issues and propose appropriate solutions to address them. Furthermore, some medical open resource datasets, such as OHSUMED, represent both classification problems, i.e. multi-class and multi-label classification problems (William Hersh and colleagues at the Oregon Health Science University, 1998) and MIMIC medical reports (Johnson et al., 2016) respectively. The massive amounts of data in both datasets make identifying medical patterns difficult to track and explain to medical experts. As a result, it would be more convenient if we could automatically explain the decision of any medical classification or prediction system.

Regarding any medical record, a doctor or physician is interested in knowing which medical entities or features are behind disease predictions or decisions. Hence, the need to create an automatic medical classification system that is more explainable and interpretable to human beings is obvious. This also means that any such medical system must be trustworthy and reliable in order to be useful in such sensitive medical decisions (Jha et al., 2018) (Angelov & Soares, 2020). Therefore, most NLP researchers are now attempting to develop automatic medical systems that are more explainable and understandable to human end-users.

Explainable AI (XAI) methods are divided into several categories based on the nature of the system they describe. So, before employing any XAI method for our system, we must first ask ourselves some fundamental questions. For example, can the XAI method interpret a specific model or can it be generalized for use on any system? Alternatively, we must determine whether our XAI method explains the system's predictions locally or globally. More importantly, a decision must be made regarding which aspect of the model needs to be explained. Three categories define the type of XAI method based on that concept: Pre-Model explanation, In-Model

explanation, and Post-Model explanation (also known as post-hoc models) (Arrieta et al., 2020)(Ribeiro et al., 2016) (Singh et al., 2020). Eventually, each explainable method must be quantitatively or qualitatively evaluated, which is a difficult but doable task.

An XAI system must be evaluated, especially when attempting to compare a newly created one to others in the literature. As a result, subjectively assessing the quality of the XAI system is dependent on the understanding of the end-user. The goal here is to satisfy the understanding of humans and to allow them to judge the effectiveness of your XAI method. Hence, there are two kinds of evaluation methods that rely on qualitative assessment. The first is known as a human-grounded assessment, and it necessitates direct interaction with the end-user, regardless of his or her knowledge of the system at hand. The second one is more practical in that it selects more domain experts to judge the explainability of your XAI method. However, both techniques are expensive in time and effort, and it may be possible to employ other practical and less time-consuming options, so using quantitative evaluation methods is more practical and efficient. Functionally grounded is an evaluation technique that doesn't need the interaction of humans to identify the quality of your XAI model (Islam et al., 2020). It deals with the criteria of some measurements that serve as proxies, such as the depth of the decision tree, model sparsity, and uncertainty. Accordingly, quantitative techniques would be more automatic and practical than the previously mentioned qualitative techniques (Carvalho et al., 2019)(Mohseni et al., 2021)(Molnar, 2019)(Van Der Waa et al., 2021).

Aside from the significant time and effort required, one of the difficulties in qualitative XAI evaluation is that requesting human opinion to subjectively assess a particular explainability method can be biased to their desire and satisfaction. As a result, quantitative evaluation techniques during the evaluation process could be more objective and time-saving. One of the main issues in the XAI research community is generalizing quantitative evaluation methods (Doshi-Velez & Kim, 2018), because dealing with customized evaluation metrics cannot be applied to all other explainability methods. As a result, we require a mutual evaluation technique that is

inclusively well suited to common explainability methods.

1.2 Motivation

The big picture of this thesis is to create an automatic and objective XAI evaluation model to evaluate the complexity of XAI method. In that context, that XAI method explains the decisions of a medical multi-class classification system that uses a combined method of two feature representation techniques: BoW and word embedding. Medical data contains so many features that lead to a particular diagnosis, disease symptom, or treatment. Accordingly, it is very important to discover medical patterns that lead to a medical decision and explain it to the end user.

OHSUMED dataset is a multi-class cardiovascular unstructured data. It is considered as a challenging data due to the large number or classes, 23 cardiovascular classes. Hence, the multi-class classificaction models suffer from tracking the medical patterns and relate them with each individual cardiovascular class. On the other hand, it is hard to explain the medical decisions of these classification models to the end users which will make the process of identifying any disease diagnosis or treatment hard to discover or track.

Feature representation techniques play an important part in any classification system. However, the traditional BoW techniques, such as TF-IDF, have a lack with the ability to capture the syntax and the sentential context of words (Kowsari et al., 2019). On the other hand, word embedding systems solved the drawbacks of BoW techniques by discovering syntax and context relations between words but the result of the WE system is simply a floating-point numerical number which gives no sense to identify important or common terms in data that any class can share (Kowsari et al., 2019) (Sinoara et al., 2019).

Additionally, the difficulty of explaining the results and decisions of the multi-class classification systems is proportional to the growing size of the features and the

classes. Hence, the necessity of using automatic ways for explainability becomes a must specially when it is related to a sensitive medical domain. Moreover, the evaluation of the existed Explainable AI methods can be done subjectively depending on the sentiment of the end user. That lead us to think more about creating objective and quantitative evaluation techniques to assess XAI methods.

Our motivation in this thesis then is to explain to the end user, i.e. in our case the doctors or any other medical party, the behavior of the multi-class classification model automatically and how it respond to this data. Following that, we will try to propose an effective quantitative evaluation technique of XAI. At the same time, we will try to enhance the performance of the multi-class classification medical system that basically relies on the representation of medical features and entities in OHSUMED dataset.

1.3 Aim of the Study

Developing an automatic and objective XAI evaluation measure is the main aim of this thesis. Thereafter, this measure will estimate the complexity of an explainable method, which is applied to a multi-class classification model as a use case. The probability of discovering a medical pattern classification model then increases by combining the two feature representation methods.

We designed several research questions that we will try to answer during the work of this dissertation:

Question 1: Does combining two feature representation techniques tackle the shortcomings of the individual representation methods and, hence, enhance the performance of the multi-class classification method?

Question 2: Which XAI method is the best to explain the behavior of the multi-class classification model and hence track the medical features that are responsible for the black-box model prediction?

Question 3: Can the depth of a decision tree be used as a quantitative complexity measurement of any XAI method?

We will answer the former research questions at the end of the results and discussion section, showing all the reasons behind the answer to each question.

1.4 Contribution of the Thesis

Each research question in the former section is considered a new contribution in this dissertation. The first contribution is the work with a challenging dataset such as OHSUMED. This dataset recorded critically low accuracy in literature (Sinoara et al., 2019) in a multi-class classification system that used a customized word embedding system. As a result, our hybrid feature representation configuration, using both BoW and WE altogether, is unique, especially in the medical domain. This technique enhanced the accuracy of our multi-class classification model from 45.4% to 50%, especially when we used PubMed and MIMIC as external data resources for the construction of the hybrid feature representation model.

For the sake of explainability, we have several contributions, which start from our experience of multiple types of XAI methods to explain the predictions of our medical classification model. The second XAI contribution is the comparison of multiple local and global XAI methods that we provide in this study. Moreover, as a third contribution, we created a quantitative XAI evaluation metric that depends on the depth of a decision tree as a proxy model. The fourth contribution is that we used SHAP XAI as a dimension reduction technique and the accuracy of our multi-class classification system incredibly increased from 50% to 92%. This method controls the most efficient and important features that are responsible for a particular prediction.

1.5 Scope of the Thesis

Our research has some boundaries that we intend to address in this section. This could be a great opportunity to take this study into the development of new XAI methods as well as other feature representation methods. The boundaries are:

1. As a result of the difficulty of finding medical data, only one benchmark was used in this study, OHSUMED data.
2. This study targeted only one domain, i.e. the medical domain. It is possible to extend this study to include other domains.
3. This study includes only explaining the predictions or the output of the black-box model, i.e. Post-model XAI. It is out of the scope of this thesis to address and explain the core functionality or the in-model behavior of the black-box.

1.6 Thesis Organization

This thesis consists of 6 chapters. The second chapter provides in details the most recent studies that are relevant to our topic. Chapter 3 describes in details the major steps of our methodology. It provides information about the dataset, the different steps to build our systems, and a comparison between the different XAI methods. Chapter 4 on the other hand shows all the involved experiments with their settings and technical environments. Chapter 5 represents our findings, results and their discussions. Then finally chapter 6 provides the summary of our work and the future plans.

CHAPTER TWO
BACKGROUND and LITERATURE REVIEW

This chapter provides two main sections. The first one presents some important definitions related to the main topics in this thesis. It starts with giving some brief definitions and information about the different feature representation techniques: BoW and WE methods. Next in the other part, we describe Explainable AI (XAI) concept and its different types. However, in each main section we provide some recent studies that are related to each topic.

2.1 BoW vs Word Embeddings

Bag of Words (BoW) is simply one of the most important feature representation methods, which relays on the fact of ignoring the order of the words. However, one can imagine putting all the words in a bag and the order of these words lost inside that bag. The most important thing now is to take the count or the co-occurrence of each word inside that bag into account. Accordingly, each word now is represented as a number that accumulates how many times the word exists in a text. As a result, the term " Term frequency" will be attached to that number regardless of the of the order of that word in the text. Another approach on the other hand can create a binary vector rather than a simple counting number. This approach tracks only whether a word exists or not inside the text. However, this method results in the diversity of the words as well as their order and grammar. The length of a document, as well as other documents inside a class, on the other hand is ignored by the Count Vector, which calculates the number of times each word in a document appears (Sunagar et al., 2020). Some noisy terms, such as sop words, can affect calculating TF negatively due to their frequently existence in the document. Therefore, it is preferable to remove stop words before calculating TF (Kowsari et al., 2019). Term Frequency-Inverse Document Frequency (TF-IDF) instead, shows how a particular term is important, by calculating their occurrences, amongst the whole corpus. This technique helps to ignore insignificant terms in data, such as stop words, and focus on the important one. As a result, both methods lack the

ability to capture the syntax and the sentential context of words (Kowsari et al., 2019).

Word Embeddings (WE) is a feature vector representation method that has the ability to solve the drawbacks of BoW techniques. WE can discover syntax and context relations between words (Kowsari et al., 2019) (Sinoara et al., 2019).The numerical vector representation is created as a result of training neural networks, by adjusting its weights, to predict a value as an output. For example, Autoencoders is considered one type of unsupervised neural networks that is responsible for taking a word as input then encode it to a numerical vector representation as an output (Ye et al., 2018). As a result, there is one drawback of WE. The output of these networks is a floating-point numerical number which gives no sense to identify important or common terms in data that any class can share (Sinoara et al., 2019). In other words, WE can give equally importance to all vocabulary in the data. For example, in a medical domain, the classifier needs to identify medical terms in the whole data rather than all vocabularies. In sentiment analysis also, a classifier needs to recognize sentimental terms. Therefore, there are some attends in literature, as we will provide in the next sections, that tries to make progress in this issue.

2.1.1 Traditional WE Systems

The best advantage of word embeddings in comparison to the other feature representation techniques, such as TF-IDF, One-Hot Encoding (Count Vectorizing), and Co-Occurrence Matrix, that it can recognize syntactic and context relationships between words. Hence the different approaches for word embeddings in literature relies on that assumption. The commonly WE systems that are frequently used in literature: Wor2Vec: (Mikolov et al., 2013) proposed a method for creating vectors by training a neural network that consists of two hidden layers. The first hidden layer is called continuous bag of words (CBOW) that predicts a target word(t) depending on its neighboring words in context word(t-1) and word(t+1). On the other hand, the second hidden layer, which is called continuous skip gram layer, is responsible for predicting the context words depending on a target word that can be found in the same

sentence. Word2Vec tends to infer the meaning of a word by the company, context words, it keeps. For example, if you have two words that have similar neighbors, i.e. similar context, then these words are quite similar in meaning. Figure 2.1 shows a simple example of how Word2Vec can predict a target word, the center word, depending on its context and window size (represented by C in Figure 2.1). Taking the sentence "Analyzing white blood cells can detect leukemia." as an example, with a target word "blood" and its neighbor words. When the window size is one (C=1), it means that it takes into consideration one word after and before the target word, such as "white" and "cells" respectively. Additionally, when the window size is equal to two (C=2), it means that it takes into account two consecutive words before and after the target word, such as "Analyzing white" and "cells can" respectively.

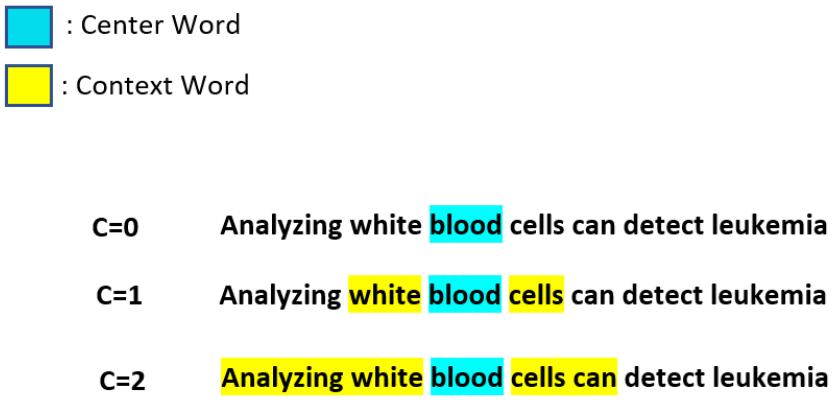


Figure 2.1 A simple example of how Word2Vec works

Global Vectors for Word Representation (Glove): This technique is represented by (Pennington et al., 2014), and it works the same way the Word2vec method work. The only difference that Glove has to train over huge data resources such as Wikipedia 2014 and Gigaword 5 with 50-dimension word vector representation. Moreover, they used Twitter data to pre-train their word embeddings to expand the word vector dimension to 100, 200 and 300 long. Glove is considered a new global log bilinear regression model that combines the advantages of the two major model families in the literature: global matrix factorization and local context window methods. Figure 2.2 shows a visualization of the word distances over a sample data set.

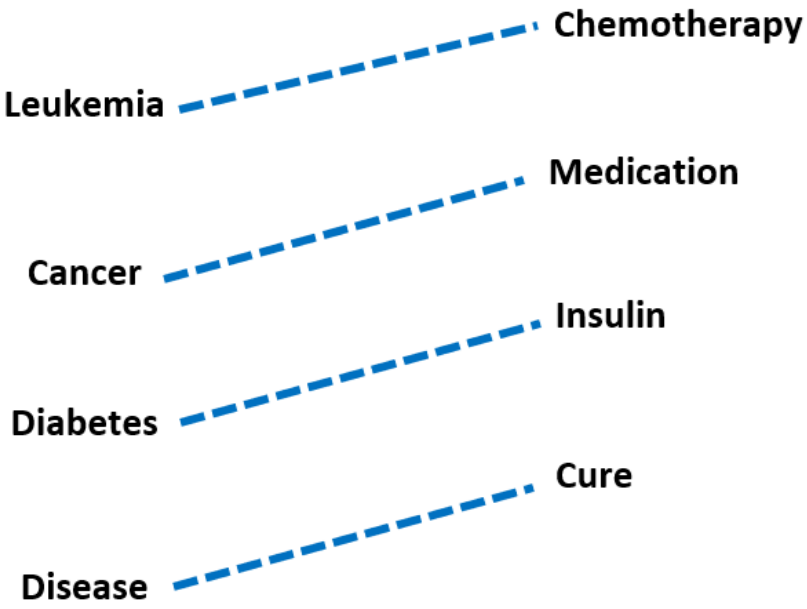


Figure 2.2 GloVe: Global Vectors for Word Representation

The example in Figure 2.2 shows how you can find a target word by taking into consideration the relation between two other related words (Words analogies). For example, the underlying concept that distinguishes Leukemia from Chemotherapy, i.e. Disease or Cure, may be equivalently specified by various other word pairs, such as Diabetes and Insulin or Cancer and Medicine.

FastText: many researchers depend on assigning a distinct word vector to morphological words. (Bojanowski et al., 2017) proposed a word embedding technique to solve the problem of word morphology. The main idea is to take into consideration that each word is represented as a bag of its *n*-gram characters. Each created vector then is related to a *n*-gram character set of a target word. For example, given the word "introduce" and $n = 3$, FastText will produce the following representation composed of character tri-grams: < in, int, ntr, tro, rod, odu, duc, uce, ce > Note that the sequence <int>, corresponding to the word here is different from the tri-gram "int" from the word "introduce ". This technique is considered fast even if it deals with larger corpora. Besides, it can give a word representation to those words which did not appear in the training dataset.

Context2vec: This technique is presented by (Melamud et al., 2016), and aims to learn a sentence-based word embedding representation taking into consideration contexts around a target word. This method depends on using a bidirectional LSTM neural network to capture context words from both directions of the target word. The output layer of that network represents the embedding of the whole sentential context beside the embedding of the target word itself. Context2vect solved many NLP problems such as Polysemy and it is capable of capturing syntax and semantics of the word.

Although the mentioned approaches solved many NLP problems, they still need to be extended and expanded to deal with different datasets and different domains.

2.1.2 Extending Word Embeddings

There are other techniques in the literature which extend the work on creating word embedding. Those systems rely on the previous basic approaches and add more techniques to solve many challenges:

Dealing with Different Data Levels: Working in a particular dataset level is one challenge of extending the work on the word embedding systems. There are five main data levels in literature: character, word, sentence, paragraph and document levels. For example, as a character level, (Bojanowski et al., 2017) and (Li et al., 2019) dealt with training n-gram character set to predict word vectors. Hence, they can solve many problems related with word morphology or out-of-vocabulary words (words that didn't appear in the training set and has no vectors). (Bojanowski et al., 2017)'s approach predicts a target word vector representation by summing the n-gram vectors of its character set so that any word doesn't appear in there training set has also its vector representation. Other research dealt with word level representation as well such as (Mikolov et al., 2013) and (Pennington et al., 2014). (Mikolov et al., 2013) created an embedding system, Word2Vec, that predict the vector of the target word depending on the context of its neighbor words and vice versa. The same thing also is

done by (Pennington et al., 2014) who built an embedding system so that each word is presented by a high dimension vector and it is trained locally, based on the surrounding words, and globally over a huge corpus. However, word level methods suffer from some problems such as polysemy, out-of-vocabulary words and morphology which we explained some of their solutions previously. Furthermore, other research worked on sentence level representation such as (Melamud et al., 2016) and (Chen et al., 2019) to capture contextual representation of each word. Also, sentence-based embedding methods can recognize syntax and semantic relations of the words, especially when it deals with the context words that surround the target word within the same sentence. On the level of paragraph and document-based, words embedding also took place in literature. For example (Sinoara et al., 2019) worked on creating a knowledge-based document-level word embeddings for text classification. His method solved many NPL problems regarding word sense disambiguation WSD, identifying semantic richness of the word, and word-sense embedded vectors.

Dealing with Different Dataset Resources: Another technique to extend the work on word embeddings is the use of different data resources. There are several data resources that can be employed to create word embeddings. Wikipedia, Google documents, Twitter, PubMed.... etc. are some examples of those resources. Additionally, some other resources, such as WordNet, can also be used as an external knowledge to support semantics to the created system (Sinoara et al., 2019). A majority of researchers used Wikipedia documents such as (Pennington et al., 2014), (Wang et al., 2018), (Bojanowski et al., 2017) and (Soliman et al., 2017) since Wikipedia provides valuable and rich information about all the applications and domains in our lives. Google news also is well-thought-out as another data resources and it has been used by (Wang et al., 2018) (Mikolov et al., 2013) for creating their embeddings. Besides, electronic clinical notes and biomedical publications such as PubMed and MIMIC III data also were used by (Chen et al., 2019),(Wang et al., 2018),and (Shen et al., 2018) to create a medical word embedding. Other data resources also have been so beneficial to create word embeddings in literature such as: Gigaword 5 (Pennington et al., 2014), Twitter (Soliman et al., 2017) (Pennington

et al., 2014) , bug reports and the source code files (Ye et al., 2016).

2.1.3 WE Systems in Medical Data Domain

Many vital studies have addressed word embeddings for biomedical text classification. For example, the Sentence2Vec word embedding system has been used to create vector features (Chen et al., 2019). This technique operates as a sentence-based WE (each vector represents an entire sentence rather than a single word). Later, the word embedding system was tested on a multiclass classification problem that relies on classifying the cancer corpus's multiclass hallmarks (Chen et al., 2019). Another study (Blanco et al., 2020) focused on developing a biomedical WE system by combining several benchmark WE systems, namely FastText, Word2Vec, and Glove. It was thought that the benefits of these WE systems could be used by combining the vectors generated by each method. The results from the combined WE system were used to classify Spanish clinical electronic health records (HER) with ICD-10 codes annotated on them. (Blanco et al., 2020). (Zhang et al., 2018), on the other hand, proposed a biomedical WE system that relies on sentence encoders to generate vectors, which were then used as a feature representation method in their multilabel classification model. Following that, they developed a convolutional residual neural network model to classify EHR cohorts collected from Duke Hospital patients. A Boltzmann machine was also used to capture and detect label dependency.

In general, all of the previously mentioned studies demonstrated that using a domain-based WE in the biomedical field can improve performance over using a non-medical WE. Accordingly, there is a lot of research being done to develop and improve current WE systems.

2.1.4 Combining WE and BoW (Hybrid Approaches)

Since improving WE by combining multiple feature representation techniques is within the scope of this study, we will only address related research here. Combining several feature representation methods can be accomplished in two ways: within WE systems or by merging two or more different feature representation techniques as a hybrid approach. For example, (Pagliardini et al., 2017) worked on developing a sentence-based WE system called Sent2Vec. The average of the vectors of words that reside within the context of a specific sentence was used in the combination process. (Le & Mikolov, 2014) , on the other hand, created a document-based WE system that mapped a single document into a single vector. The resultant vectors of all the words in a given document were averaged or concatenated to form one unique vector. Several studies, however, have combined two or more different feature representation techniques, such as WE and BoW. (Enríquez et al., 2016) developed a sentiment classification system by combining WE and BoW. The combination method was based on a simple voting system that produced a confidence value for both feature representation techniques. Since the resultant values are between zero and one, they calculated an average of the values from both techniques. On the other hand (Hu et al., 2018) They developed a recommendation system to find similar bug reports. Their system aimed to use four different vector representations, or similarity scores, from four different techniques. The first score took into account the BoW method, while the second took into account WE. Bug product and component created the third similarity score, which focused on the structural relationship between two bugs. Finally, the fourth similarity score was determined by the document-level latent relationship between two bugs. As a result, the final score was calculated using the combination technique = (score1 + score2 + score4) * score3. At that point, the system recommended the k bug reports that were most similar to a given query bug. (Schmidt, 2019) They attempted to improve their work on word embedding systems by combining them with some TF-IDF weighting techniques. They used several weighting techniques, including term frequency (TF), inverse document frequency (IDF), and smooth inverse frequency, along with a Word2vec model's sub-sampling

function. The created word embedding vectors, however, were implemented at the document level, which aggregated all relevant word vectors. The sum of each weighting technique for a specific word with its related word embedding vector was then computed. According to their findings, the IDF weighting technique outperformed the others in terms of ROC and AUC performance metrics. (Liu et al., 2018) represented yet another weighted approach based on BoW and WE. The combination technique begins by determining the weights of each term in the corpus. Their weighting strategy was based on two widely used BoW methods, TF-IDF and class probability. The frequency of each term within a given class is referred to as class probability. They assumed that by adding class probability information to TF-IDF weights, their model could recognize the importance and/or uniqueness of terms in a specific class domain. They then used a simple multiplication operation to add up the weights associated with each term vector. (Zhou et al., 2019) conducted another useful study in which they combined three NLP approaches, namely BoW, topic modeling, and word embeddings. They first computed TF-IDF scores for each term within a given document, and then used an LDA topic modeling approach to represent topics with terms. Then, for each term in the corpus, they extracted word embedding vectors. They attempted to combine the previously mentioned approaches in two ways. The first weighting approach combined a specific word's TF-IDF scores with its related word vector, and then the updated vectors were joined at the end with LDA scores, whereas the second weighting approach combined TF-IDF and WE as a first step. Ultimately, all of the weighted feature representation techniques mentioned above demonstrated their effectiveness in improving the performance of some ML problems.

2.2 Explainable AI (XAI)

During the last decade people in ML community try their best to make their models more explainable to human. Hence, each one tries to justify the behavior of their models properly. Moreover, one may say that Explainability and interpretebility

are two sides of one coin. However, (Luckey et al., 2022) attempted to provide definitions for both terms as well as an explanation of how each term fits into the overall XAI framework. In his opinion, Explainability can be regarded of as meta data supplied to ML algorithms, such as features that contribute to the output of ML algorithms, generated either by external algorithms or by ML algorithms themselves. On the other hand, interpretation aims to establish a domain-specific interpretation of the meta information identified in the XAI explanation part, such as by linking the most important features to structural dynamical mechanical values. To put it another way, an interpretation is the transformation of an abstract idea (such as a predicted class) into a domain that humans can comprehend. While an explanation is a set of interpretable domain features that have contributed to a particular problem in the production of a decision (e.g. classification or regression).

2.2.1 Traditional XAI Methods

ML Models That Has Explainability Nature: The simplest way to investigate model explainability is to use a model that is explainable, i.e., interpretable outside of the black box. Models with regular parameters like linear regression, logistic regression, tree-based models, rule-fits, and even k-nearest neighbors and Naive Bayes are typically included (Molnar, 2019). Explainable models have to fulfill the following main properties (Molnar, 2019):

Model Linearity: that interprets the linear relationship between the features and the model's target output.

Model Monotonicity: A monotonic model has to guarantee that the association between a feature and the target output is always in one stable direction (increase or decrease) over the feature (in its entirety of its range of values).

Model Interactions: you can always add interactional features to your model specially if your model has non-linear nature with manual feature engineering. This can be achieved manually or automatically.

Feature Importance XAI Methods: Feature importance investigates the extent to which each predictive model can behave in relation to a specific feature. Skater, which measures the entropy in the prediction change given a perturbation of a specific feature, is one of several frameworks used in the literature. Another approach is the SHAP framework, which combines feature representation with game theory. The global feature importance is then calculated by taking the average of the SHAP value magnitudes across the entire dataset (Molnar, 2019) (Pavlov, 2019). Furthermore, one of the most well-known and widely used feature importance techniques in the literature is Local Interpretable Model-agnostic Explanations (LIME) (Molnar, 2019) (Ribeiro et al., 2016). To locally explain a black-box model, LIME employs an explainable model as a surrogate (typically, a model with an interpretable nature, such as linear models, trees, or rule-based models).

Partial Dependence Plots (PDP): This method computes the margin influence of a specific feature on model prediction while leaving other features in the same model constant. PDP can also determine whether the relationship between a specific feature and a target is linear, monotonic, or more complex (Molnar, 2019).

Global Surrogate Models: A global surrogate model is an explainable model that is trained on all the predictions of a black box model, i.e., global decision. Assume you have a black box model, such as a specific type of neural network. Then you should explain this model and how its decisions were made. As a result, you can train on the neural network's predictions using an ML model with an explainable nature, such as a decision tree. Surrogate models, as a result, do not require any information about the internal mechanisms of the black box model; only the relationship between input and predicted output is used (Molnar, 2019).

2.2.2 XAI of Word Embedding Systems

The structure and dimensions of Word embeddings are complex and difficult to understand even for experts. For that purpose, some researchers tried to work on

explaining word embedding systems. There are two types of WE explainability. The first one aims to transform the original unexplained WE vector to a form suitable for others to understand. And the another aims to create an explainable WE system from scratch. (Şenel et al., 2018) tried to explain their original Glove WE system by transforming the vectors to be more understandable to beings. They believe that a semantic concept can be identified by a particular WE dimension. Hence, they assumed also a particular WE dimension can only represent a semantic category if the average value of that dimension for all the semantic category words is above a specific threshold. Therefore, they proposed two transformation matrices, one that uses the mean as input weights to the original WE systems. And the second uses a distance metric, called Bhattacharya distance, to replace the mean method. After using some quantitative and qualitative assessment to two methods, they found that the applied explainable WE methods outperform some of the state-of-the-art WE systems.

(Jha et al., 2018) similarly tried to improve the explainability of the WE in medical domain by transforming the original WE into an explainable one. The basic idea is to create a transformation matrix that is sensitive to some external, categorical, and medical knowledge. They believe that these categories are closer to the human understanding. The first step is to identify categorical terms from their corpus. After that, they created a dictionary definition for each categorical term. Then, they specified all the medical concepts in these definitions. Moreover, they expand the number of concepts by using some external resources to add their synonyms. As a result, each category term is represented as a collection of multiple medical concepts. Each one of these concepts has its own WE embedding vector from Word2Vec. They used a categorical loss function to augment the dimensions of each dictionary concept with its related category term. According to their quantitative and qualitative evaluation, this transformed form of WE became more explainable than the original one.

(Panigrahi et al., 2019) alternatively created an explainable WE system that depends on the different senses of the word. Basically, they established two different ideas: one that relays on identifying the different senses of a particular word. And the

second uses contextual information of that word besides its senses. To clarify more, the first idea depends mainly on creating word embedding vectors in such a way that each dimension will represent one sense of that word. Accordingly, they used Latent Dirichlet Allocation LDA model to define all the senses, denoted as w, of a word w. After that, they put all the w to create the dimensions of the new WE system. This technique registered a good performance in comparison with the state-of-the-art WE systems such as Word2Vec in only two over six NLP tasks. To that end, they tried to improve the performance of the current Word2sence by adding some contextual information and call it WordCtx2Sence.

As we saw in the former articles, they suffer from obvious drawbacks when it comes to explaining WE systems. On one hand, the first and second articles attempted to transform an original WE vector to another explainable space, which does not directly explain the dimensions of the original WE vector. On the other hand, the third article undertook a new explainable WE system; however, they struggle with a little trade-off in performance compared with state-of-art WE systems like Word2Vec.

2.2.3 Explainable Methods on ML Predictions

Several articles dealt with explaining black-box models in a post-operation manner. Therefore, some researchers tried to explain the predictions of the black-box model using feature importance techniques. (Godin et al., 2018) for example used a feature importance method to investigate which patterns a neural network followed, after learning character-level features, to explain a word-level tagging and output of their neural network . On the other hand, (Ge et al., 2018) used a feature importance weighting approach in the medical domain. They distinguished which top 10 features are responsible for predicting intensive care unit (ICU) mortality. Moreover, (Ribeiro et al., 2016) created a feature importance technique that depends on the local model-agnostic method (LIME) . LIME produces explanations that depend on estimating the model together with a "locally faithful" explainable representation.We can also use the nature of some linear and explainable models, as proxy models, to

interpret the predictions of a black-box model. LIME (Ribeiro et al., 2016) learns local-based surrogate models that depend on input perturbation, as we mentioned before, to produce explanations. Furthermore, (Blanco et al., 2020) proposed an explainable surrogate model that depends on Microaggregation . Microaggregation derives explanations from the black-box model's decisions, while controlling their accuracy, and compares it with a decision tree as a surrogate model.

2.2.4 Explainable AI in the Medical Domain

Some studies showed the importance of applying Explainable methods to medical image analysis. (Singh et al., 2020) mentioned two main explainability groups that are used to explain deep neural networks in the medical images: attribution-based methods, and architecture or domain-specific methods . Attribution-based methods aim to discover which input features are affecting the target Deep Neural Network (DNN) neuron or the output neuron that is responsible for identifying the correct class in a classification problem. While Domain-specific methods, sometimes called non-attribution-based methods, aim to develop an explainability method to a given problem rather than using pre-existing attribution-based methods.On the other hand, Soares's paper is a recent study that tries to explain the behavior of its classification model and to identify COVID-19 disease (Soares, 2020). Its dataset consists of CT scan images of patients infected with COVID-19. It used prototype-based learning in which a prototype, training images with a highly representative local peak of density and probability distribution, is used to explain the most important areas in the CT scans. The Explainablity of medical domain text classification also took a place in literature. The main aim here is to extract and identify features from the medical text that contribute the most to define the black-box predictions' decisions (Singh et al., 2021). (Feng et al., 2020) created an explainable clinical decision support text classifier. Their model consists of a CNN transformer that is responsible for extracting medical patient features from MIMIC III clinical records. Then, all the medical decisions are identified to be transformed later, by a two-layer transformer encoder, to recognize medical patterns among the features. (Teng et al., 2020)

additionally built an interpretable ICD-9 classification system that aims to extract information from a knowledge graph. Their system consists of several parts: a multi-layer CNN, a graph-based representation, an attention matching layer, and adversarial learning. Essentially, the knowledge graph has been constructed after applying some medical entity extraction techniques. This knowledge then is used in the attention layer to encode all the relevant medical features that contribute to identifying inter-dependency between ICD-9 codes from the MIMIC III dataset.

2.2.5 Evaluation of Explainable Methods

Quantitative explainability measurement has played a great part in evaluating explainability recently. Researchers have begun to design and perform some practical and quantitative evaluation metrics well suited to their explainability methods. (Messalas et al., 2020) tried to prove that the SHAP interpretability method is a faster alternative than LIME . Then they evaluated the performance of both techniques using six quantitative evaluation methods: Runtime, Consistency guarantees, Identity, Expressive power, Translucency, and Portability. All of those metrics depend on the features' correlations and importance. Similarly, (Ramon et al., 2020) made a comparison between three explainability methods: SHAP, LIME, and Search for Evidence Counterfactuals (SEDC) . After that, they evaluated the effectiveness, which is represented by Switching point and Percentage Explained metrics, and the efficiency, which is represented by the computational time, of the three explainable models. They found that SEDC consistently shows the best performance amongst them, while SHAP and LIME recorded the second and third best performance, respectively. (Lin et al., 2019) alternatively assessed the performance of three explainability methods: SHAP, LIME, and Expected Gradients EG . They tested these methods on three classification problems: image, text, and speech recognition using deep CNN classifiers. After that, they measured the performance of the interpretable methods by assessing the Impact on Decisions, Impact Score IS, and by assessing the erroneous coverage, Impact Coverage IC. Additionally, (Antwarg et al., 2019) compared SHAP and LIME after explaining the detection of anomalies by

autoencoders. To assess the explainability methods, they depended on the Mean Reciprocal Rank (MRR) measure to identify the noise position in the feature set. As a result, SHAP recorded a better MRR score in comparison to LIME. Furthermore, (ElShawi et al., 2020) compared six explainability methods after applying them to the predictions of a random forest black-box model . They utilized LIME, SHAP, Anchors, LORE, ILIME, and MAPLE XAI methods on healthcare tabular and textual datasets. Then they evaluated those methods using six quantitative assessment methods: similarity, time, trust, identity, stability, and separability. They concluded that SHAP performed the best in terms of time, similarity, and identity on the tabular dataset. Moreover, LIME recorded the best separability on the tabular and textual dataset and the best similarity score in the textual dataset. Last but not least, (Jesus et al., 2021) performed a subjective (i.e. qualitative) assessment built on an application-based explainability evaluation metric . They constructed three basic questions and allowed domain experts to rate SHAP and LIME from one to five stars accordingly. These questions assessed how the XAI method could cover all relevant information on a decision, and whether XAI was useful in making that decision. Consequently, they found that people are more receptive to rate SHAP better than LIME in a fraud detection problem using the random forest as a black-box model.

2.2.6 XAI As a Dimensionality Reduction Method

There are some dimensionality reduction techniques that are used in deep and machine learning applications such as Principal Component Analysis (PCA), Random Projection (RP) and Non-Negative Matrix Factorization (NMF) (Monica & Parvathi, 2019). These methods were utilized to reduce the amount of irrelevance in the data. However, the features and attributes derived from these algorithms were unable to classify data into separate divisions. Moreover, the mentioned algorithms were uncappable of solving multiclass classification problems and challenges (Kumar et al., 2018). Accordingly, we will provide two studies that used XAI as a dimensionality reduction in some classification problems.

(Dunn et al., 2021) studied the possibility of using an XAI method as a feature selection technique. More specifically, they employed SHAP XAI method as a feature selection method. For each feature of each data point, their technique assigns SHAP values, which are contribution values for a model's output. They used the contribution information of each feature to arrange the features based on their importance using these SHAP values, which represent the priority that a model gives to a feature. They calculated the mean of each feature in a class-based manner. For each SHAP matrix, the mean of each feature is calculated, which is assigned later to each class. After that, the vectors are summed together. To convey feature importance, the summed vectors are arranged in descending order. Finally, only the top m feature will be kept for the classification task.

(Marcílio & Eler, 2020) similarly made a comparison of using several feature importance techniques as a feature selection method. Across a series of experiments, they compare the capacity of CART, Optimal Trees, XGBoost, and SHAP to accurately identify the relevant subset of data features. The results reveal that XGBoost fails to discriminate between significant and irrelevant features regardless of whether they employ the other native feature importance techniques or SHAP.

As a result, XAI is still a new field and researchers are working on solving explainability problems specially on discovering new XAI evaluation techniques. However, the work on explainability in the medical field is still limited and needs more research to accomplish (Lim et al., 2021). On the other hand, making comparisons between the various XAI approaches is also important, as it reveals which technique is more appropriate to utilize in a specific domain or dataset. Accordingly, we tried to tackle most of XAI's challenges and problems by creating a generic automated and objective XAI evaluation method. In addition, we provided a comparison between two benchmark XAI methods in a medical domain problem.

CHAPTER THREE
METHODOLOGY

This chapter provides a detailed description of our complete XAI system. It is basically consists of multiple stages and parts that together serve as one solid and integrated XAI system. However, this thesis consists of several studies that represents each main section in this chapter. Therefore, and according to Figure 3.1, we will denote each study in the experimentation chapter as the following:

1. Study 1: The study starts with the creation of our hybrid feature representation method. This stage basically maintains the collection of two types of data. The first one is used to produce our WE and BoW systems, such as PubMed and MIMIC III data, and the other one is used to evaluate our system, such as OHSUMED data. After that we explain the methods, which are used for combining both feature representation techniques. Then we explain the steps and parts of the multi-class LSTM neural network that is employed as a classifier for the multi-class OHSUMED data classification.

2. Study 2: In this part, which is called the XAI stage, we try to apply several XAI techniques on the predictions of the former mentioned system. This step is considered as a comparison stage of several XAI methods that have been utilized in the literature.

3. Study 3: In this step, we describe our new method of evaluating explainable AI quantitatively. This stage will describe which XAI has the advantage to be used as a trustworthy XAI method.

4. Study 4: Then finally in the last stage, we try to enhance the accuracy of our multi-class classification system by employing SHAP XAI method as a dimensionality reduction technique on OHSUMED dataset.

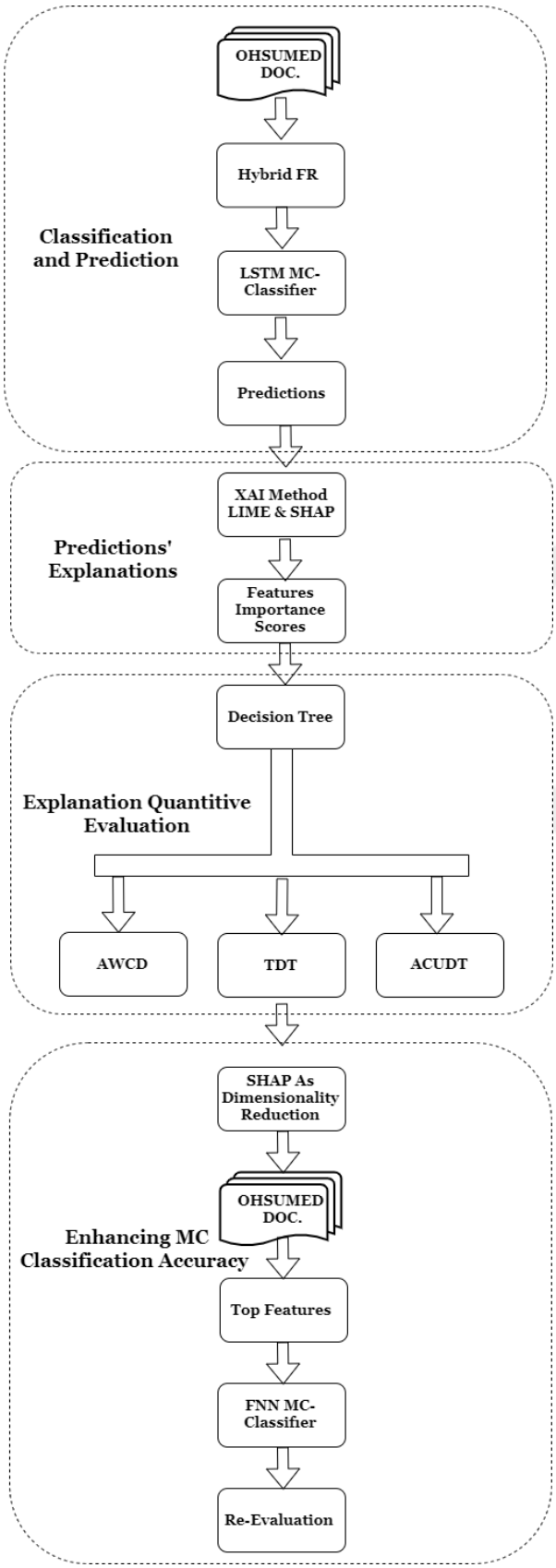


Figure 3.1 General Architecture of the XAI System

3.1 Classification of multi-class Data Using LSTM Neural Network and a Weighted Hybrid Feature Representation Method

Each strategy we suggested in our methodology is described in depth in this section. Our approach first explains how we built a weighting feature representation system by collecting the data and then using a subset of that data to compute word-embedding vectors. Additionally, it provides information on how to calculate the weights for each word and their combinations for term frequency (TF), inverse document frequency (IDF), and class probability (CP). The multi-class classification model we developed to evaluate our weighting strategy is then describe as following.

3.1.1 Datasets' Description and Preprocessing

We used three different types of biomedical datasets from the literature. We developed the model with two of them, PubMed and MIMIC-III, using our weighting feature representation approach. The third dataset is the OHSUMED dataset, a multi-class dataset employed to test our proposed weighting feature representation system. Each of the three datasets is described briefly below:

1. PubMed is one of the most well-known biomedical databases that provides access to the MEDLINE biomedical database, which contains research papers and is widely used for citation tracking. The PubMed database is 170GB in size (as of 27 January 2020 statistics) and contains approximately 32 million biomedical articles, abstracts, bibliometrics, citations, and related data, all of which are stored and updated on a regular basis in our local repository. Many researchers in biomedical studies who are looking for reliable evidence-based biomedical resources consider PubMed to be a benchmark for any biomedical applications that use machine learning models (Chen et al., 2019)(Gargiulo et al., 2019). Due to computer limitations, we were only able to work with two million PubMed abstracts rather than the entire collection. Table 3.1 contains

information about the collection we used.

2. MIMIC III, which stands for Biomedical Information Mart for Intensive Care, contains over 40,000 patient documents from the intensive care units at Beth Israel Deaconess Medical Center between 2001 and 2012 (Johnson et al., 2016). Furthermore, the dataset includes 53,423 adult hospital admissions and 7870 neonatal records. As a result, this database contains over 20 million clinical summaries. In our case, we used all the existing 20 million textual clinical summaries for patients. This dataset is described briefly in Table 3.1.

3. OHSUMED is a subset of medical abstracts extracted from the MEDLINE database (William Hersh and colleagues at the Oregon Health Science University, 1998). William Hersh and colleagues at Oregon Health Science University launched and organized this dataset (Tan, 2007). Each document in this set is associated with one of the 23 cardiovascular disease categories. The total number of documents in the collected dataset is 13,929 in total. We under-sampled the data to make it more balanced and convenient for the multi-classification system because the collected documents were highly unbalanced. As a result, each of the 23 cardiovascular diseases was associated with 400 documents. Finally, we used 9200 abstracts in total, 70% for training and 30% for testing. Table 3.1 summarizes all of the pertinent details about the OHSUMED dataset.

Table 3.1 Key information about the three datasets used in this study.

Descriptions	PubMed	MIMIC III	OHSUMED
Total no. of documents	2,000,000	2,083,180	9200
Total no. of sentences	13,218,036	40,378,672	76,863
Total no. of tokens	300,577,868	606,768,871	1,658,785
Total no. of characters	1,959,234,200	3,458,104,347	10,797,237
Max. document size in sentence	62	559	23
Max. document size in tokens	1624	8770	605
Max. document size in characters	9954	55,293	4025
Avg. document size in sentence	6	19	2
Avg. document size in token	150	291	180
Avg. document size in characters	980	1660	1173
Total no. of classes	N/A	N/A	23

3.1.2 Data Prepocessing

In this step, we used data preprocessing techniques and libraries to remove unwanted characters and punctuation marks from our datasets. We identified and cleaned the dataset with the NLTK Python libraries. We used the following operations to accomplish this:

- The quotation mark was removed.
- The numbers were removed.
- All characters were converted to lowercase.
- Tokenization of words.

3.1.3 Weighted Feature Representation

This step is divided into three sub-steps. The first explains how we determined the TF, IDF, and CP values for each word in the corpus. The second step shows how we developed our word embedding vectors. The third step demonstrates how we combined the previous two methods to create a single improved feature representation.

To clarify, suppose we have a dataset D with mtext documents to classify, $D = (d_1, d_2, d_3,..., d_m)$, where an arbitrary document d_i is represented by a set of terms. $d_i = (t_1, t_2, t_3,..., t_n)$, and each term t_i can be represented by a single weighting score, such as term frequency score (TF), inverse document frequency (IDF), or class probability (CP). In our case, w_j denotes one of the previously mentioned weighting techniques of the term t_j in the document $d_i = <w_1, w_2, w_3,..., w_n>$. The entire set of terms in dataset D constitutes a vocabulary of text documents, $V = (t_1, t_2, t_3,..., t_n)$, where an arbitrary term t_i is represented by a feature set of F_i, which contains two distinct features (weighted value and word embedding vectors). $F_i = (s_i, we_i)$ where s_i and we_i are the *ith* term's weighting score and word embedding vectors, respectively.

3.1.3.1 Calculation of Weighting Techniques

We calculated all of the weighting scores for each distinct word in the evaluation OHSUMED dataset for this step. For calculating the TF and IDF scores for all vocabulary in the OHSUMED-400 dataset, we relied on Formulas (3.1) and (3.2) (Schmidt, 2019). Class Probability CP was calculated using Formula (3.3) (Liu et al., 2018).

TF_{t_j} is a function that returns the term frequency in document d_i normalized by the number of terms. This weighting technique demonstrates how important a specific term t_i is based on its co-occurrence in document d_i. As a result, we want to see how this technique improves classification performance when combined with the word embedding vector.

$$TF_{t_j} = \frac{f_{t_i,d_j}}{|d_i|} \tag{3.1}$$

Where f_{t_i,d_j} represents the number of times the term t_j appears in a document d_i, and $|d_i|$ represents the document's size in terms of terms. IDF_{t_j} is the inverse document frequency of term t_j and can be defined as in Equation (3.2).

$$IDF_{t_j} = log\frac{N}{m} \tag{3.2}$$

Where N is the number of documents that contain the term t_j and m is the total number of documents in the dataset D. The IDF weighting technique represents how frequently or infrequently a specific term t_i appears in the corpus. More common words, such as stop words, will have lower IDF weights than rare words. As a result, this technique determines the importance of rare and domain-specific words in our corpus, such as medical terms.

Class probability P($t_j|C_k$) or CP_{t_j,C_k} is the conditional probability that shows how many times the term t_j appears when the class C_k appears. CP demonstrates the significance of a specific termt_i while taking into account class information. For example, if the term t_i appears multiple times in documents annotated as class A, it indicates that the term is closely related to that class.

$$CP_{t_j,C_k} = \frac{t_j \bigcap C_k}{P(C_k)} \tag{3.3}$$

where P(C_k) denotes the probability of class C_k. As a result, three weighted scores are assigned to each term t_j in the dataset d_i:

$$s_j = \sum_{d=0}^{m} s_j = (TF_{t_j}, IDF_{t_j}, CP_{t_j}) \tag{3.4}$$

Each word in the vocabulary is now associated with one of the three weighted

values s_j. For example, if the document d_2 only contains three words, V= (t_1^{leukemia},t_2^{blood},$t_3^{\text{antiglobulin}}$) s_j score values are as follows:

s_1= (TF^{leukemia} = 0.75, IDF^{leukemia} = 0.66, CP^{leukemia} = 0.99)

s_2= (TF^{blood} = 0.18, IDF^{blood} = 0.36, CP^{blood} = 0.56)

s_3= ($TF^{\text{antiglobulin}}$ = 0.59, $IDF^{\text{antiglobulin}}$ = 0.98, $CP^{\text{antiglobulin}}$ = 0.43)

3.1.3.2 Creation of Word Embedding (WE) Vectors

We used PubMed and MIMIC III, with a total of four million documents, to create a word embedding vector. For each word in the vocabulary, we used the FastText WE library (Bojanowski et al., 2017) to compute a 100-dimensional vector space. FastText generates a powerful model that has solved numerous natural language problems, such as word morphology and out-of-vocabulary word problems (Kowsari et al., 2019). As a result, we constructed word vectors from the four million documents using the Gensim Python library. Therefore, we had a vocabulary V that included all of the distinct terms in the dataset D, $V = (t_1, t_2, t_3,..., t_n)$, with each term represented by a 100-dimensional WE vector.

Assuming we have only three words in V, the resulting 100-dimensional word embedding vector, (we), for each term in V is defined as follows after running the FastText library to extract the feature:

we_1^{leukemia}= <-3.215408, -4.233652,..., 0.2386677>

we_2^{blood}= <-6.8077517, 7.012626,..., 3.39976>

$we_3^{\text{antiglobulin}}$= <1.4542218, 0.642345,..., -3.6465673>

The values of we vectors are the actual values from our training models for each term.

3.1.3.3 Weighted Word Embedding Vectors

In this section, we show how to combine WE vectors and weighting scores to update the structure of the most recent or original WE vector form. This means that we used a simple weighting factor to combine WE vectors and the three weighting scores for each term t_i in the vocabulary V.

Let t_j represent a specific word in the vocabulary V. Then, for an arbitrary term t_j associated with three weighting scores in (s_j) and each item of its related WE vector (we_j), the weighted multiplication can be performed as follows, yielding three weighted feature vectors, HTF_j, $HIDF_j$ and HCP_j, respectively, of the three weighting techniques TF, IDF, and CP.

Term frequency (TF) is a weighting method. It simply multiplies each term's TF score t_j by each item in its (we_j) vector.

$$HTF_j = \sum_{l=0}^{m} TF_{t_j} \times we_{j,l} \tag{3.5}$$

For example:

$HTF_{\text{leukemia}} = TF_{\text{leukemia}} \times we_{\text{leukemia},l}$

$HTF_{\text{leukemia}} = (0.1108 \times (-3.215), 0.1108 \times (-4.234), ..., 0.1108 \times (0.239))$

$HTF_{\text{leukemia}} = (-0.356108, -0.468879, ..., 0.026432)$

Alternatively, IDF and class probability (CP) scores can be used as weights in the same manner:

$$HIDF_j = \sum_{l=0}^{m} IDF_{t_j} \times we_{j,l} \tag{3.6}$$

$$HCP_j = \sum_{l=0}^{m} CP_{t_j} \times we_{j,l} \tag{3.7}$$

3.1.4 Multi-class Classification Model

After getting the combined vector of features, it was necessary to evaluate it in a case study, including a multi-class classification task. As inputs, we utilized the OHSUMED dataset (Tan, 2007) and the vector of features, then projected one output over 23 classes for each test data set. First, we developed a model for multi-class categorization based on a bidirectional long-short-term memory (BLSTM). The BLSTM model can recognize patterns and interpret texts in both directions. However, this model requires a longer learning period than LSTM or recurrent neural network (RNN) models. This model is comprised of the following layers, which are depicted in further detail in Figure 3.3:

The input layer: this layer feeds each document (a series of words) to the neural network as input.

The embedding layer: this layer provides a mapping between every word in the input text and the embedding list. At that moment, a floating point vector is retrieved for each word. However, the floating-point value may be derived from the training dataset or from any external information that has been pretrained. Thus, we utilized knowledge derived from the PubMed and MIMIC III databases. Consequently, our input data is represented as feature vectors.

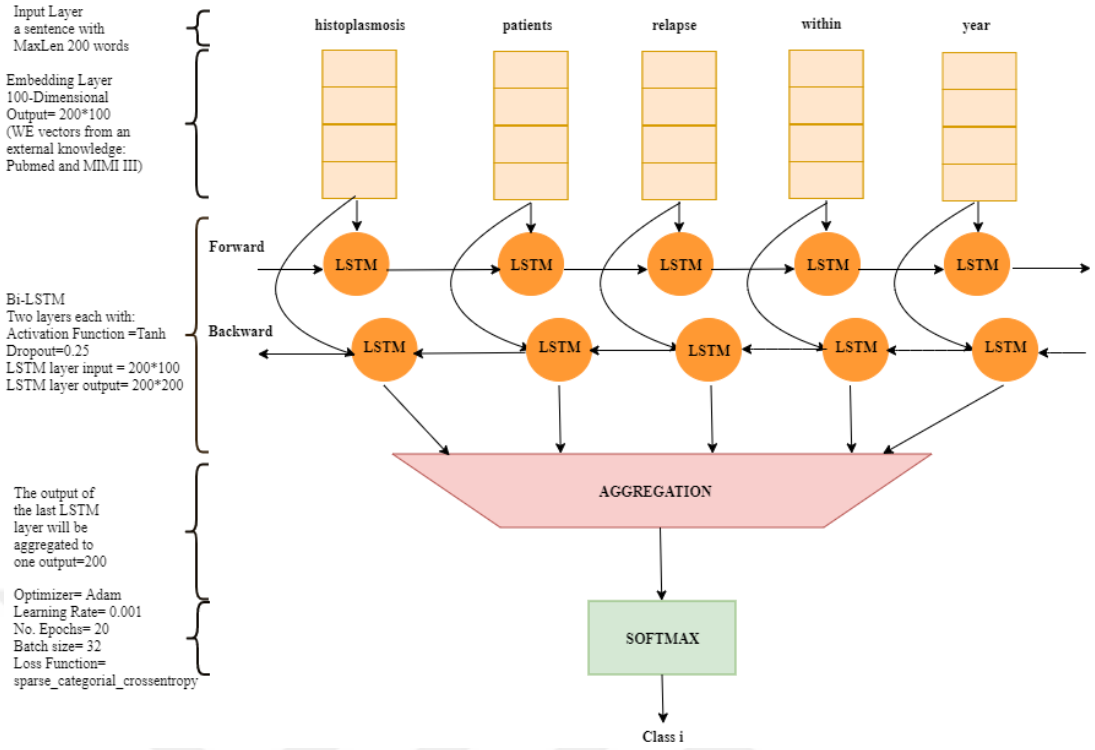


Figure 3.2 The architecture of the multi-class classification system.

BLSTM layers: we built one BLSTM layer with the Tanh hyperparameters for the activation function and a dropout value of 0.25.

The output layer: the responsibility of this layer is to forecast the output class. Since this is a multi-class model, it must predict one of the 23 categories of cardiovascular diseases. SoftMax was utilized as the activation function for this layer because it is suitable for multi-class problems.

In addition, we empirically tuned key hyperparameters that simplified the learning process and yielded positive outcomes. We selected the Adam optimizer, set the learning rate to 0.001 and the decay value to 1×10^{-6} for the sparse-categorical-crossentropy loss function with 20 epochs and 32 batches.

3.2 Explianable AI (XAI) on the Predictions of the Multi-class Classification System

As we explained earlier in the introduction about the types of XAI methods, we decide to work on post-model explainability. It means that we worked on explaining the output of the black-box model embodied with multi-class classification predictions. As a result, we tried to explain the predictions and decisions of our multi-class classification model using local and global XAI methods. The local explainability methods are represented by feature importance techniques such as LIME, SHAP, and ELI5. While the global techniques are represented by three surrogate models and a model with an explainability nature such as a decision tree. The aim of this section is to provide a comparison of different XAI methods. Figure 3.4 shows the different XAI methods that were used in this study.

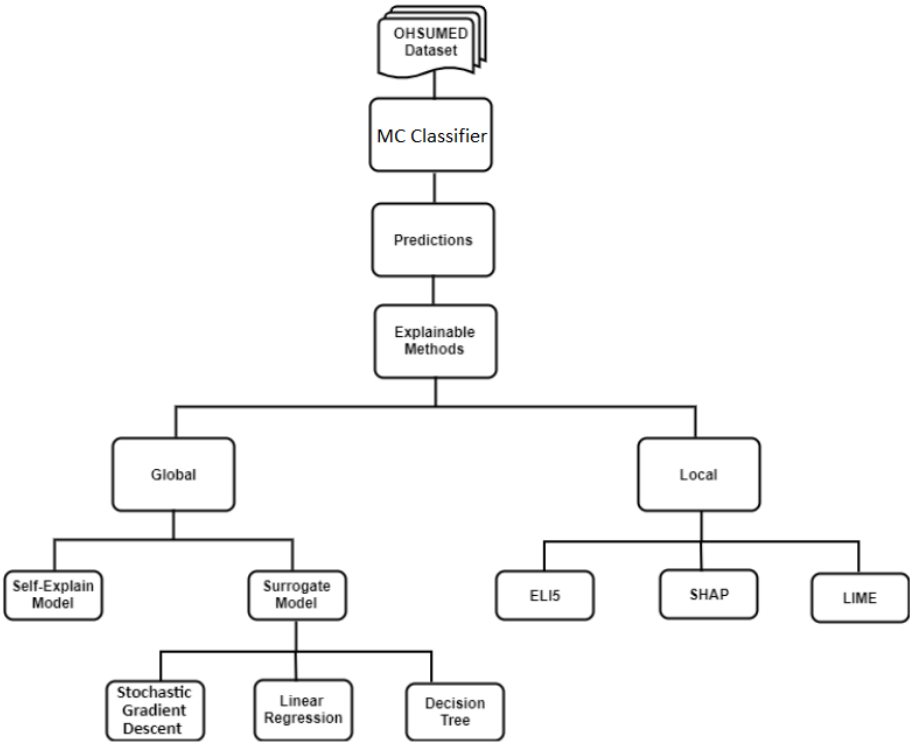


Figure 3.3 XAI methods used for explaining the multi-class classifier

3.2.1 Feature Importance Method

In general, feature importance is defined as the degree to which features are related to the black-box model's prediction decisions. Intuitively, the more the model's decisions are based on the feature, the more sensitive its predictions become to changes in that feature. We can assess the significance of a feature by first calculating the increase in error after perturbing the feature. Second, a feature is considered important if the model's error increases proportionally to the perturbing value of that feature. That is, the model's predictions are extremely sensitive to that feature, and thus it is deemed "important" (Molnar, 2019). In this study we tested three famous feature importance techniques on the predictions of our multi-class classification model.

3.2.1.1 LIME

Local Interpretable Model-agnostic Explanations (LIME) is one of the most famous feature importance techniques that is used in the literature (Ribeiro et al., 2016) (Molnar, 2019). LIME uses an explainable model as a surrogate (usually, a model that has interpretable nature such as linear models, trees, or rule-based models) to locally explain a black-box model. Its work principle starts with selecting an instance of interest, a document, whose explanation is desired after obtaining the black-box prediction. Secondly, LIME perturbs or adds noise to the dataset and obtains the black-box predictions on the disturbed data. Then, LIME weighs new data samples by calculating their closeness to the instance of interest. After that, it fits a weighted interpretable model, the surrogate model, on the perturbed dataset. And finally, LIME explains the black-box predictions by interpreting the local model.

3.2.1.2 SHAP

SHapley Additive exPlanations (SHAP), on the other hand, connects game theory principles with local explanations to represent the only possible consistent local features attributed with the black-box decisions (Molnar, 2019) (Lundberg & Lee, 2017). The predictions of the black-box model can be explained by assuming that each feature is a "player" in a game, and each prediction is considered as the "payout" of that game. The Shapley value, which is a big factor in the game theory perspective, represents how to fairly assign the "payout", or predictions, to each "player" or feature. SHAP's working principles starts with representing a prediction problem of a single data instance as a "game". Then, the "gain" or "payout" of the game is the actual instance predictions subtracted by the average predictions of all the other instances. And finally, the "players" are the features that were collaborated together to receive the "gain".

3.2.1.3 ELI5

Explain Like I'm a 5 (ELI5) is the easiest explainable technique that depends also in features perturbation to get the importance of the features. ELI5 is a python library package that is good to start with to simply explain a black-box model. ELI5 supports tree-based, linear and parametric models but doesn't support model-agnostic interpretations (Molnar, 2019).

3.2.1.4 Global Surrogate Models

All the above-mentioned techniques are good to explain the behaviour of a black-box model from a local instance point of view. But what if we want to get the big picture of the total behaviour of our black-box model? A global surrogate model builds an approximated estimation of a black-box model to be closer to an interpretable model. Using a proxy model is considered a model-agnostic method that

doesn't require information about the inner behaviour of the black-box model. It just uses its interpretable behaviour to explain the decisions and the results of the black-box model(Molnar, 2019). The following scenario explains how global surrogate models work:

1. For a particular dataset in hand, get the predictions of the black-box model accordingly.
2. Select a particular interpretable mode, such as linear, tree-based or rule-based models, to work as a surrogate.
3. Train your surrogate model on the predictions of the black-box model.
4. Interpret or visualize the decision of the surrogate model which actually depends on the predictions of the black-box model.

3.2.1.5 A Model with Explainability Nature

There are some machine learning models that contains a global explainable nature. You can use them directly and globally to explain the predictions and decisions but at the expense of general performance. For example, the explainability of a decision tree and some other linear based models is greater than some other black-box models such as neural networks. However, the dilemma in this scenario is that the performance of these interpretable models in some cases is much lower than those in neural networks. Hence, if you care more about explainability regardless of the performance or the accuracy of the model, you can use interpretable models directly instead of using a mysterious black-box model (Molnar, 2019). In this study, we explored three explainable models, such as Linear Regression, Stochastic Gradient Decent and a tree-based ML model, to directly explain the global behaviour of the predictions' decisions.

3.3 XAI Evaluation System

In this section, we propose a novel and more practical method for quantitatively evaluating a specific XAI method. For comparison purposes, we chose the work with SHAP and LIME XAI methods because they are model agnostic and are extensively and frequently applied to a wide range of health-care data and industry domain to interpret deep learning models (Zhou et al., 2021) (Hailemariam et al., 2020) (Ghassemi et al., 2021). However, our main goal in this section is to create an automated evaluation metric that effectively and objectively measures XAI complexity. As a result, we chose to measure XAI complexity using the depth of the decision tree because it is a practical and straightforward method (Arrieta et al., 2020). Decision trees are one of the few machine learning algorithms that a human can easily track and analyze (Hearty & Gibney, 2008). Hence, it will add simplicity and reliability when it will be used as an XAI evaluation measurement. Additionally, we intended to prove that our method can alleviate various issues related to human-based evaluation techniques and generalization problems.

Accordingly, our proposal begins with using the depth of the tree to evaluate an XAI's complexity by feeding its feature importance scores into the tree. As a result, the depth of the decision tree can indicate how complex the XAI method is (Molnar, 2019) (Carvalho et al., 2019) (Zhou et al., 2021). To be more specific, we used our neural network classification system to classify the multi-class cardiovascular dataset (OHSUMED). Then, to explain our black-box model's decisions, we focused on used using two cutting-edge explainability methods: SHapley Additive exPlanations Sheply (SHAP) and Local Interpretable Model-agnostic Explanations (LIME). The depth of the decision tree was then used to assess the complexity of the explanation methods.

Accordingly, in the XAI phase, we calculate the importance score of each feature in the OHSUMED dataset using Algorithm 1. As a results, each feature f_m inside each document d_i now is attached with one importance score s_m. A matrix X then will be used to represent the output of this process. Each element or raw in X now represents

a number of importance scores that map each raw in the OHSUMED dataset.

Algorithm 1 The calculation of LIME and SHAP XAI importance scores

Input: OHSUMED documents D = (d_1, d_2, d_3.....d_n) each related to one class c_i where C = (c_1, c_2, c_3....c_n).

Output: A matrix X contains: a set of documents d_n represented by its features f_m and its related LIME/SHAP importance scores s_m. X=(s_1, s_2, s_3....., s_m)

```
for d_i in D do
    for f_m in d_i do
        X←Calculate LIME/SHAP importance score s_m.
    end for
end for
```

3.3.1 A Quantitative Evaluation Technique of Explainability

Each feature in the previous section is associated with a single importance score from a specific explainability method. As a result, we employ the depth of the decision tree (DT) as a metric for measuring the complexity of the classification system. Instead of using the original sequences of OHSUMED documents, we use the explainability model's feature importance scores as an input to the decision tree. Then we let the DT classify the OHSUMED documents based on the importance of each feature. Following that, we use the resulting rules and graph to determine the depth of the tree and thus whether or not the explainability technique is complex. Algorithm 2 shows how we calculate each XAI evaluation metric.

For example, the tree in Figure 3.4 is the result of feeding XAI feature importance scores into a decision tree-based classifier. The root of the tree contains the entire set of data used in the classification problem, which in this case is four samples. Following that, each branch stores information about each node, such as the number of samples in each node and the Gini score. Additionally, this tree has four leaves, each of which represents a class or the target of a specific decision. In our case, the four classes are

C_{13}, C_{06}, C_{19}, and C_{22}. As a result, this tree can hold three different XAI evaluation measurements, two for XAI complexity and one for DT performance. Thus, the total depth of the decision tree is the first complexity measurement (TDT). The second is the average of the weighted class depth (AWCD). Finally, the accuracy of the DT classification model (ACUDT) is used to evaluate the decision tree's performance.

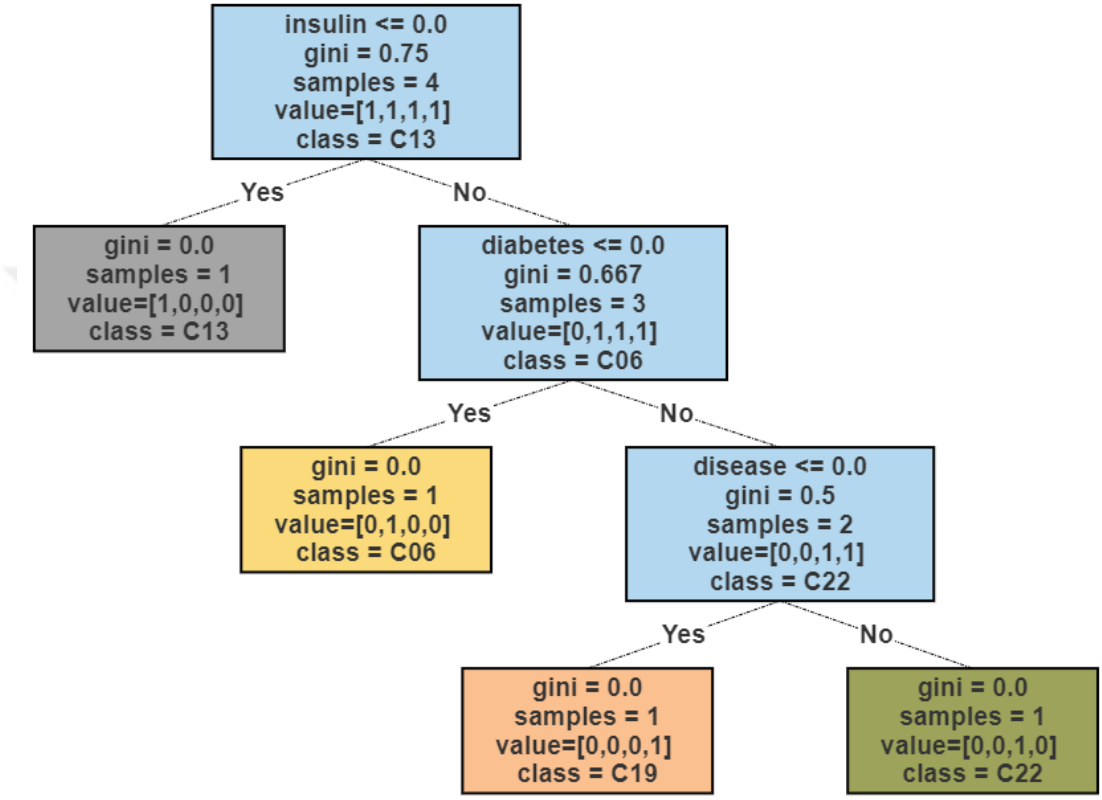


Figure 3.4 A decision tree graph of four document samples.

3.3.1.1 The Total Depth of the DT

Some researchers have suggested that the total depth of a decision tree can be used as a complexity indicator (Molnar, 2019) (Carvalho et al., 2019) (Zhou et al., 2021). The model becomes more complex as the depth of the DT increases. Hence, the total depth of the tree in Figure 3.4 is:

$$TDT = Count(L) \tag{3.8}$$

Where L represents each level of the tree; the root is not included. Consequently, the total number of tree levels TDT in the Figure 3.6 tree is three. Equation 3.8 and algorithm 2 represent how TDT is calculated.

3.3.1.2 The Average of the Weighted Class Depth

Instead of the total depth of the tree, this measurement can tell us about the complexity in a class-based depth. As previously stated, this study comprises 23 cardiovascular classes. In the resulting decision tree, each class represents a leaf. Therefore, each leaf can be found at a specific level. Class C_{13}, for example, exists in the DT's first level. In level two, class C_{06} is a leaf. C_{19} and C_{22} are in the tree's level three. In this case, we compute the average of the weighted (by number of samples) class depth (AWCD) as follows:

$$AWCD = \frac{1}{N} \sum_{L=1}^{l} L_{class} \times C_S \tag{3.9}$$

Where N is the total number of data samples, in this case four in Figure 3.4. L is the total number of leaves in the tree; the tree uses only four classes in total. Each class leaf's level is represented by L_{class}. Hence, each class leaf, C_{13}, C_{06}, C_{19}, and C_{22} is in the first, second, and third levels, respectively. Finally, the number of data samples in the current leaf is denoted by C_S. In that case, the number of samples in each class leaf, C_{13}, C_{06}, C_{19}, and C_{22}, is equal to one. As a result, Equation 3.9 can be mathematically calculated as follows:

$AWCD = \frac{1}{N} \times [[L_{C13} \times Saples_{C13}] + [L_{C06} \times Saples_{C06}] + [L_{C19} \times Saples_{C19}] + [L_{C22} \times Saples_{C22}]]$

$AWCD = \frac{1}{4} \times [[1 \times 1] + [2 \times 1] + [3 \times 1] + [3 \times 1]]$

$AWCD = \frac{9}{4} = 2.25$

3.3.1.3 The Accuracy of DT

We can also consider the DT classifier's accuracy (ACUDT) to represent how the DT performs when XAI's feature importance scores are used as inputs to the DT. In that case, we can rely on the DT's ability to correctly classify the documents. Furthermore, this metric can be used to determine which features will improve the performance of the black-box classifier.

Algorithm 2 The calculation of each XAI evaluation metric

Input: The matrix X, which is resulted from Algorithm 1, where each raw is attached with its related class c_i.

Output: The three XAI evaluation metrics.

1: Apply a Decision Tree (DT) algorithm to the matrix X and its related class c_i.
2: Draw DT that consists of multiple levels L= (l_1, l_2, l_3...,l_z), and its leaves as c_i.
3: TDT ← Count (L).
4: **for** each class c_i and its related number of data samples $samp_s$ in a particular DT level Lc_i **do**
5: AWCD ← $\frac{1}{X}\sum_{L}^{1} Lc_i \times \text{samp}_s$
6: **end for**
7: ACUDT ← DT accuracy score.

3.4 XAI As a Dimensionality Reduction Method

Multi-class problem presents a number of difficulties, particularly when it comes to medical issues. However, OHSMED dataset is multi-class dataset in which each document is related to one over 23 cardiovascular diseases. As a result, there are numerous difficulties with this dataset due to the large number of classes and interrelationships between classes and features. Accordingly, these limitations make finding patterns to create a prediction considerably more difficult. This section

focuses on solving the problem of misidentification of the correct classes in OHSUMED dataset. Accordingly, we propose a new dimensionality reduction depending on the feature importance scores of SHAP explainability XAI method. We believed that SHAP XAI method has the ability of assigning importance scores to features so that we can select the most significant and relevant features for each document of OHSUMED.

3.4.1 Using SHAP Feature Importance Scores As a Dimensionality Reduction Method

(Marcílio & Eler, 2020) used SHAP as a feature selection method but in a class-based manner. It means that they only select the top features of each class group of documents. Unfortunately, by applying that, there will be a loss of information by selecting a set of features that maybe important for a particular document within that class, but also irrelevant for another document within the same class. Accordingly, our approach will be done in a document-based way. The steps in algorithm 3 show how we select the top features that are relevant for each document using SHAP XAI method. See the last part of Fig 3.1.

Algorithm 3 Using SHAP as a dimensionality reduction method

Input: OHSUMED documents D = (d_1, d_2, d_3.....d_n) each related to one class c_i where C = (c_1, c_2, c_3....c_n).

Output: A matrix X contains: a set of documents d_n represented by its top t of features f_m and its related SHAP importance scores s_m. X_t=(s_1, s_2, s_3....., s_t) that is considered in the multi-class classification

1: **for** d_i in D **do**
2: **for** f_m in d_i **do**
3: X←Calculate SHAP importance score s_m.
4: Arrange f_m in a descending order according to their SHAP importance scores s_m.
5: Select the top t features and consider them in the classification model.
6: **end for**
7: **end for**

3.4.2 The Multi-class Classification System

In this stage, we preferred to apply a simple Feed Foreword Neural Network (FNN) instead of the complex LSTM neural network. We believe that the complex structure of LSTM neural network, as it is described in the first section, was one reason among others to degrade the accuracy of the multi-class classification. In LSTM neural network, the number of parameters and weights were increased rapidly, which resulted with high false positive and negative during the prediction process. As a result, we decided to work with a simple neural network architecture that is suitable with the entered data from the previous section. Accordingly,our multi-class classification model used in this stage is a simple FNN with one input layer, one hidden layer, and one output layer. Moreover, we established several hyper-parameters that provide us with high accuracy and low loss scores:

- Activation Function in the input and hidden layers: Relu.

- Activation Function in the output layer: Softmax.
- Loss Function: Sparse-categorical-crossentropy.
- Optimizer: Adam.
- Epochs =15, 20 and 30.
- Batch-size= 5.

At the end, our system is a complete XAI system that regardless of solving XAI problems and challenges, it also aims to solve multi-class problems. The results chapter will show in details the progress that we made to build and enhance an effective XAI system.

CHAPTER FOUR
EXPERIMENTS

This chapter describes in details all the experiments involved in this dissertation. Each section here will explain the experimental set up for each part of the former discussed methodology as follows:

1. The experiments applied on the "Classification of Multi-class Data Using BLSTM Neural Network and a Weighted Hybrid Feature Representation Method" study. Denoted by Study 1 for short.
2. The experiments applied on the "Explianable AI (XAI) on the Predictions of the Multi-class Classification System" study. Denoted by Study 2 for short.
3. The experiments applied on the "XAI Evaluation System" study. Denoted by Study 3 for short.
4. The experiments applied on the "XAI As a Dimensionality Reduction Method" study. Denoted by Study 4 for short.

4.1 Evaluation Metrics

We examined our performance outcomes with respect to two generally employed measurement criteria, namely training accuracy and validation accuracy. Many neural network applications are familiar with these metrics.

Accuracy is our model's capacity to accurately categorize documents. Nevertheless, there are two common forms of accuracy in neural networks: training accuracy and validation accuracy (Ahmed et al., 2019). Training accuracy is our model's capacity to accurately categorize data during the training phase. Validation accuracy indicates the quality and precision of our model during testing and validation. Both are computed in every neural network epoch. In contrast, the loss metric computes the error in the classification model's prediction that might occur during the fine-tuning of the neural

network's weights (Ahmed et al., 2019). Consequently, there are two forms of this measure associated with the training and validation procedures.

4.2 Experiments on Study 1

To assess and compare our method presented in section 3.1, we ran a series of experiments and examined the performance results of the BLSTM deep learning architecture in comparison to other architectures and machine learning techniques. In this part, we also describe the performance metrics used to evaluate our model.

4.2.1 Experimentation Setup

When we attempted to manage the PubMed and MIMIC-III databases, we ran into a number of issues. Our low RAM and processor resources were incapable of managing the massive amount of data. We utilized the Google Colab and TRUBA infrastructures as our primary hardware resources to solve this issue. We were, at that time, originally introduced to Google Colab. Its tensor processing unit (TPU) has over 100 GB of RAM and 30 GB of storage space. This infrastructure performed well for running experiments to validate our model, but when we attempted to utilize the same environment to construct the FastText WE vector, it was unable to finish the task. Instead, we chose TRUBA, a computer cluster system supplied by the Turkish National Science Foundation (TÜB ITAK) that includes high-performance computing gear and data storage facilities. We employed a server with 128 nodes, each with eight cores of two Xeon E5-2690 2.90 GHz processors and 256 GB of RAM. For constructing our neural network, we used Python 3.6 together with the Keras and TensorFlow frameworks. In addition, the Gensim Python module was utilized to generate both the FastText WE vector and the weighted scores. The only purpose of the primary trials was to establish and fix the hyperparameter values of our BLSTM model. We experimented with a number of hyper-parameters, including dropout levels between 0.1 and 0.5, and then examined other activation functions, including

Tanh, Sigmoid, and Relu.

Then, we explored with epochs values between 10 and 32 and batch sizes between 10 and 32. We experimented with optimizers like ADAM, SGD, and RMS-prop. As a consequence, we adjusted all BLSTM hyper-parameters following Section 3.1.4. We conducted all tests for multi-class classification of biomedical literature using the 9200-document OHSUMED dataset. Recognizing that we were dealing with a multi-class system, we deleted all duplicate documents from the dataset after determining that certain documents had several classes. Therefore, the number of remaining papers became 7512. The remaining materials were divided into two categories: 80% for training (6,010 items) and 20% for testing (1502 documents). As a result, each document was only assigned one of 23 cardiovascular disease groups. As a baseline, the first experiment was conducted on the original WE vector (i.e., the WE prior to the combination operation). We performed this to determine how the weighted feature representation methods could alter the classification model's performance compared to the original WE. The same environment and hyper-parameters were then applied to the HTF, HIDF, and HCP hybrid feature vectors. We also conducted additional trials on certain cutting-edge machine learning techniques and methods. We evaluated our initial feature vectors at this step.

As a result, we chose a set of ML algorithms, including logistic regression (LR),k-nearest neighbors (k-NN)(withk=5), random forest (RF), support vector machine (SVM), linear support vector machine (LSVM),decision tree (DT), multi-layer perceptron (MLP), and Gaussian naïve based (GNB) classification. In these experiments, we regarded their hyperparameters' default values. In addition, we compared our model to one of the most relevant studies in the literature. In that article (Sinoara et al., 2019), the authors evaluated their word-embedding-based feature representation technique using the same OHSUMED data we utilized in this investigation. In actuality, we evaluated our model using the identical performance metrics indicated in their report. Although comparable word-embedding-based modeling was developed, the model was not particularly trained for the biological domain. Utilizing the same dataset as their research enables us to employ it for

comparative reasons.

4.3 Experiments on Study 2

We used a computer with core i7, RAM 8 GB and windows 10 operating system to perform these experiments. In addition, we used Python 3.6 and the spider platform to write our code. Each prediction from BLSTM model has been explained locally and globally using the following python libraries:

- Local XAI SHAP: Using shap library.
- Local XAI LIME: Using lime library.
- Local XAI ELI5: Using eli5.TextExplainer library.
- Global Surrogate XAI Decision Tree : Using interpret.ext.glassbox.DecisionTreeExplainableModel library.
- Global Surrogate XAI Linear Regression : Using interpret.ext.glassbox.LinearExplainableModel library.
- Global Surrogate XAI Stochastic Gradient Descent : Using interpret.ext.glassbox.SGDExplainableModel library.
- Self-Explain Model Decision Tree : Using sklearn.tree.DecisionTreeClassifier library.

However, the aim of Study 2 is to provide a comparison of different XAI methods. Hence, in the result section, we will provide a comparison of all the former mentioned XAI methods in terms of visual explainability showing in our perspective all the cons and pros of each XAI method.

4.4 Experiments on Study 3

4.4.1 Experimentation Setup

In the methodology section, we provided all the information on our neural network's hyper-parameters. In addition, we used Python 3.6 and the spider platform to write our code. You can find the full code of the system in our GitHub repository https://github.com/nizar3112/DT_XAI_EVALUATION.git.We performed several experiments to investigate the following:

1. Based on explainability feature importance scores, how DT accuracy scales with the number of documents.
2. How the number of top feature sets with the highest importance score affects the DT accuracy.
3. Whether the number of features and/or documents affect the total depth of the DT.
4. How the number of features and/or documents directly impacts the average of the weighted class depth of the DT.

4.4.2 LIME and SHAP Experiments

To obtain the explanations for each document, we used the LIME and SHAP libraries, which are compatible with our neural network infrastructure. As a result, each document is now represented by a series of feature importance scores derived from the LIME and SHAP XAI methods. However, we set up several experiments in the following manner:

1. Document-based experiments: for each XAI method, we set up four experiments based on the number of documents. We assigned 100 documents (baseline), 500

documents, 1400 documents (only test documents), and the entire set of 6900 documents (train and test sets). This parameter, on the other hand, can show us how the accuracy and depth of DT are affected on a document basis.

2. Experiments with the top feature set: we began to audit the number of top features with the highest importance scores as a dynamic parameter. This parameter can help us understand how to adjust the accuracy and DT depth. Accordingly, we set that parameter to 20, 50, 100, and 500, the maximum number of features that a document can have.

3. Experiments to enhance the accuracy of the multi-class classification model using SHAP as a dimensionality reduction method: for each document in OHSUMED dataset, we choose the top 100, 200, 300, and 400 features after calculating their SHAP importance scores in the multi-class classification. This step was responsible for increasing the accuracy of our black-box model especially when the number of epochs increased to 30.

4.5 Experiments on Study 4

4.5.1 Experimentation Setup

We evaluate our multi-class classification model using the accuracy metric. Additionally, we worked with 10-fold cross validation to ensure the stability of our model. We worked in a windows 10 environment of i7 intel microprocessor. Additionally, the experiments took place in python IDE spider with several deep learning and XAI libraries such as: KERAS, Sklearn and SHAP libraries respectively.

We prepared four experiments depending on each set of features, i.e., 100, 200,300 and 400 features. After that, we observed which feature set will record the highest accuracy. We devised four experiments, one for each set of features, namely 100, 200, 300, and 400. Then we looked at which feature set would record the highest accuracy.

CHAPTER FIVE
RESULTS and DISCUSSION

In this part we'll go through all of the findings from the experimentation mentioned in the previous chapter. Accordingly, we have four results sections, each of which is related with a single study, i.e. from study 1 to 4, which were maintained in the method and experiment chapters.

5.1 Results and Discussion of Study 1

In this part, we cover all the training and validation accuracy and loss findings. In addition, we provide the outcomes of our BLSTM model using both the original baseline FastText WE system and the weighted feature representation strategies employing the aforementioned metric values. In addition, the results of the experiments are presented in order to give a comparison with other regularly used ML methods for multi-class research.

5.1.1 BLSTM Multi-class Classification Results

According to the validation accuracy and loss in Table 5.1, the CP weighting approach had the maximum performance by achieving 49.4% and 1.901, respectively. We also realize that the validation accuracy difference between CP and the two weighting procedures, TF and IDF, was only 0.01 percentage points. In contrast, the baseline data demonstrated the best training accuracy and loss among all of them. In terms of validation accuracy, all weighting strategies performed better than the baseline in the final analysis.

Table 5.1 The results of our model with each feature vector method.

Metrics	**Baseline FastText WE**	**TF Weighted**	**IDF Weighted**	**CP Weighted**
Training Accuracy in %	**96.2**	69.0	72.4	68.0
Validation Accuracy in %	45.4	48.6	48/3	**49.4**
Training Loss	**0.149**	1.336	0.899	1.036
Validation Loss	3.011	1.966	2.018	**1.901**

5.1.2 Results of Other ML Approaches

Table 5.2 demonstrates that the linear SVM had the second-best performance among ML algorithms in terms of accuracy score. On the other hand, the original FastText WE system did not respond well to decision trees. In general, the original FastText WE method generated the greatest score using our BLSTM model compared to the other ML algorithms.

Table 5.2 The accuracy of applying FastText WE with ML models.

Machine Learning Approach	**Accuracy in %**
Deep Neural Network (DNN) with BLSTM	**45.4**
Logistic Regression	37.7
k-NN (k = 5)	27.9
Random Forest	27.1
Multilayer Perceptron	32.1
Gaussian Naive Bayes	30.9
SVM	38.3
SVM (Linear kernel)	40.5
Decision Tree	12.4

5.1.3 Discussions and Comparisons of the Results

In the previous section, the CP weighting method gave the best result among all the feature weighting methods when applied to the BLSTM model. However, the difference between CP and the two other weighting schemes was just approximately 1%. In addition, all weighting approaches reported greater performance ratings compared to the baseline. To explain why these weighting techniques were effective, we must examine the working concept of the WE systems in depth. WE typically generates a feature vector for a letter, word, or even a full document. It can find syntactic and contextual relationships between words ((Kowsari et al., 2019)(Sinoara et al., 2019). Each integer in the vector provides a relevant description of the word it corresponds to. Therefore, we believe that modifying this representation may have a positive or negative effect on the classification system's performance. In our situation, the weighting methods functioned positively by including statistical terms, TF, IDF, and CP, into the WE vector. In addition, CP achieved the highest performance, which demonstrates how including information about the classes may assist in improving accuracy.

During testing of the baseline model, we discovered that giving the parameter Trainable = True or False to the embedding layer had a significant impact on the neural network's performance. Setting this parameter to True would build the embedding vectors using the training data, which in our instance, was the OHSUMED dataset. Thus, the quantity of OHSUMED documents was insufficient for constructing vectors, which might result in an over-fitting issue. In addition, if we utilized this parameter as equal to True with our external information, the structure of the vectors derived from the external dataset would be altered, which would reduce the model's precision. Therefore, we gave False to this parameter for our baseline studies. This indicates that utilizing external knowledge enhances and improves the neural network's capacity for learning.

We attempted to reduce over-fitting issues as much as possible by incorporating regularizations into our model, such as the dropout hyper-parameter and dataset

shuffling prior to splitting. As a result, the training accuracy of the baseline was greater than that of the other weighting methods, i.e. about 96.1%. Accordingly, we chose to evaluate the performance of our model based on validation accuracy.

The impact of label interdependence on the overall performance of a classification system is significant. In a multi-label classification task, this phenomena improves the system's overall performance (Tehrani & Ahrens, 2017) (Zhang et al., 2015) (He et al., 2020). In this situation, it is necessary to enhance the likelihood that many classes may coexist in a single document, but label dependency is seen as a curse for some multi-class situations, such as the OHSUMED dataset (Lei et al., 2019). Each document in the OHSUMED collection is associated with a single class. In addition, this data covers 23 kinds of cardiovascular diseases, i.e., 23 distinct diseases and conditions associated with the heart and blood vessels. This complicates the classification process by making it more difficult to recognize the unique patterns of each class, which implies that classes with the same disease conditions may share common characteristics. Therefore, sharing characteristics between classes indicates that they are interdependent. As a result, the general performance in all situations using the OHSUMED dataset was low, i.e. the scores were not highly significant. Our model was negatively affected by the enormous number of classes in the classification phase. As the number of classes involved in the classification process increased, so did the difficulty of accurately identifying each class.

In addition, we compared our findings to equivalent research in literature that developed a customized WE system and its application to the OHSUMED-400 multiclass dataset. To this end, (Sinoara et al., 2019) tested a WE model with multiple ML models, including linear SVM, NB, decision tree, and k-NN, where each model achieved validation accuracies of 38.2%, 27.9%, 10.1%, and 30.7%, respectively. Consequently, our model outperformed theirs using our WE approach with BLSTM and linear SVM. Figure 5.1 shows a comparison of our scores that are related to applying our WE vectors to BLSTM and ML algorithims with the highest score of (Sinoara et al., 2019) study.

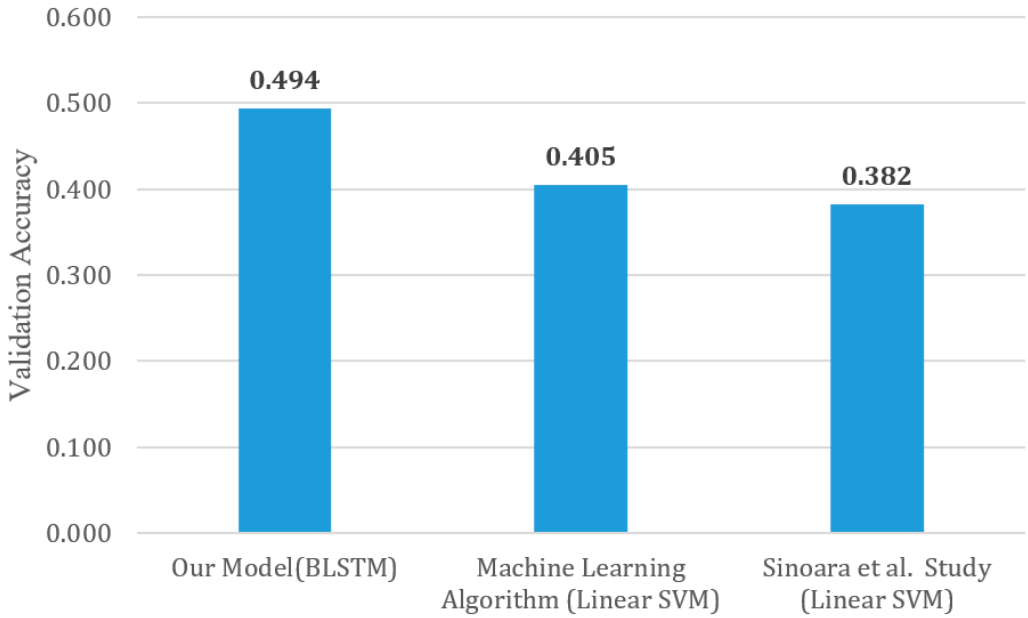


Figure 5.1 A comparison between our model and other algorithms.

5.2 Results and Discussion of Study 2

This study aims to provide a comparison between different explainable methods that were applied on our black-box model predictions. As a result, we will show the visualization results of each XAI method maintained in the experiment section. After that, at the end of this section, we will give our opinion as a comparison of the different XAI methods, showing their pros and cons during the explanation process.

5.2.1 Local Feature Importance Methods' Visualization

To show how each local XAI method works and visualized, we explained the local methods in this study by experimenting the XAI methods on one test document, i.e., one local instance.The test document has a true class = 21:C03, which means that 21 is the index of the test document and correctly classified to C03 by our black-box model.

5.2.1.1 LIME

LIME provides several types of explanations depending on how features are represented. We choose to show the most 2 clear explanations. For example, Figure 5.2 represents the features and their importance in a dictionary-like form. Furthermore, Figure 5.2 shows the top 2 classes, C03 and C08, that contribute the most during the prediction decision. We also noticed that LIME depends on the distribution of probability to priorities classes taking into account each class with its feature importance scores. The words in red notifications are the important features that contribute positively to identify the classes. For the prediction of class C03, the most important feature is "leishmaniasis" with an importance score = 0.18767221976666376.

```
Document id: 2
Predicted class = 21:C03
True class: 21:C03
--------------------------------------------------------------------

Explanation for class 15:C08
('nasal', 0.10700929224038297)
('culture', 0.03552819344680423)
('therapy', 0.029568452781541146)
('country', 0.029401879217971726)
('recommended', 0.029262711884477894)
('62', 0.02792397133095386)
('INTERVENTION', -0.02520809742026206)
('mucosa', 0.023786107409400445)
('Open', -0.02317085974466227)
('body', -0.023098753713059698)
--------------------------------------------------------------------

Explanation for class 21:C03
('leishmaniasis', 0.18767221976666376)
('antimony', 0.1651795942659426)
('patients', -0.09060619014494899)
('toxicity', -0.07985446235249263)
('mucosa', -0.06978081711915149)
('lesions', 0.04439528491092745)
('parasites', 0.04207695503817168)
('stibogluconate', 0.032737444919813646)
('antileishmanial', 0.029452511843606668)
('Leishmania', 0.02528431768529536)
```

Figure 5.2 A dictionary like feature importance scores using LIME.

On the other hand, Figure 5.3 is a more explainable form of LIME that relay on visualizing the most important features with their scores to interpret the predictions. However, this form also can provide the distribution of the important features in the original text. And we can also recognize the inter-dependency between classes accordingly. For example, our test document shows that the true class is C03, but it

also says that class C08 has some related features that existed in the same document. Consequently, C03 and C08 are inter-dependent classes.

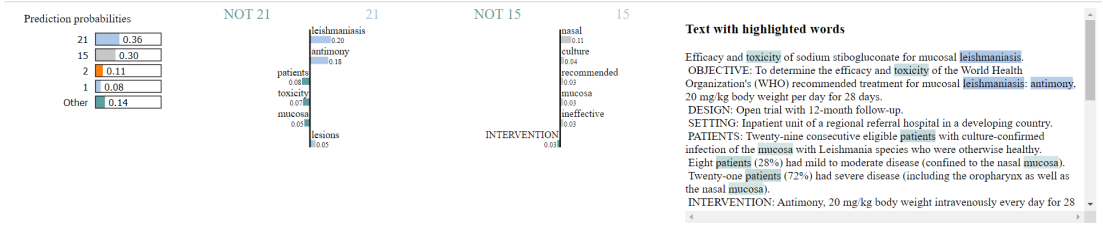


Figure 5.3 A complete visualization of LIME XAI method.

5.2.1.2 SHAP

SHAP uses a game theory principle, accompanied by local features, that are contributed to the black-box model predictions. SHAP uses Shapley scores to represent the importance of the features, or players, to identify the predictions or in that case the gain of the game. Figure 5.4 Shows the most important features that contributed to the neural network decisions.

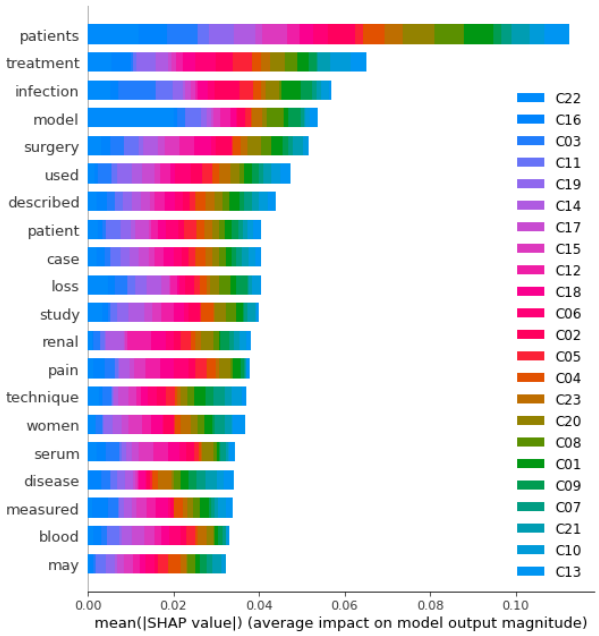


Figure 5.4 Explanations of SHAP method.

According to Figure 5.4, The top feature, that is highly important, is the word

"patients" with an importance score of more than 0.10. the different colors in the figure represent all the different classes. Accordingly, SHAP also shows the percentage of importance among different features which also distributed between classes. Like LIME, SHAP can provide inter-dependency relationships between classes as a consequence of feature intersections.

5.2.1.3 ELI5

ELI5 shows the explanations of the black-box in a simple way that is more readable by the human. Figure 5.5, for example, shows which information that ELI5 use for the explanations:

1. It shows all the classes, in our case the 23 classes, with their probabilities amongst the neural network predictions and decisions.
2. It shows the total importance score of each class.
3. And each feature's importance score individually. This score could be positive or negative.

y=18 (probability **0.011**, score **-4.030**) top features		y=21 (probability **0.020**, score **-3.364**) top features		y=0 (probability **0.013**, score **-3.817**) top features		y=22 (probability **0.186**, score **-0.818**) top features		y=1 (probability **0.031**, score **-2.922**) top features		y=15 (probability **0.022**, score **-3.278**) top features		y=8 (probability **0.089**, score **-1.762**) top features		y=17 (probability **0.013**, score **-3.832**) top features		y=20 (probability **0.006**, score **-4.605**) top features	
Contribution[?]	Feature	Contribution[?]	Feature	Contribution[?]	Feature	Contribution[?]	Feature	Contribution[?]	Feature	Contribution[?]	Feature	Contribution[?]	Feature	Contribution[?]	Feature	Contribution[?]	Feature
+0.476	primary biliary	+0.809	primary biliary	+0.819	primary biliary	+0.343	cirrhosis	+0.427	autoantibodies to	+0.566	biliary cirrhosis	+0.782	biliary cirrhosis	+0.651	biliary cirrhosis	+1.102	primary biliary
+0.417	the presence	+0.425	of autoantibodies	+0.638	biliary cirrhosis	+0.243	the mitochondrial	+0.346	mitochondrial	+0.536	presence of	+0.482	s syndrome	+0.647	sjogren s	+0.983	biliary cirrhosis
+0.414	presence of	+0.408	directed at	+0.398	the mitochondrial	+0.215	in primary		antigens	+0.460	sjogren s	+0.477	autoantibodies to	+0.570	s syndrome	+0.925	sjogren s
+0.306	96 patients	+0.402	s syndrome	+0.380	autoantibodies to	+0.197	sjogren s	+0.329	presence of	+0.387	mitochondrial	+0.292	directed at	+0.443	the presence	+0.413	have been
+0.290	sjogren s	+0.281	biliary cirrhosis	+0.300	directed at	+0.167	biliary	+0.304	for the		antigens	+0.292	primary biliary	+0.427	the mitochondrial	+0.379	presence of
+0.223	patients with	+0.265	the mitochondrial	+0.298	the presence	+0.161	s syndrome	+0.302	patients with	+0.371	s syndrome	+0.226	with sjogren	+0.425	autoantibodies to	+0.327	patients with
+0.204	cirrhosis we	+0.251	for the	+0.238	sjogren s	+0.134	reported to	+0.301	of autoantibodies	+0.365	the presence	+0.218	antigens however	+0.396	primary biliary	+0.326	for the
+0.194	and to	+0.240	sjogren s	+0.216	issue we	+0.132	issue we	+0.296	have been	+0.342	of autoantibodies	+0.212	primary	+0.369	have been	+0.320	the presence
+0.190	these latter	+0.239	96 patients	+0.210	of autoantibodies	+0.120	risk for	+0.239	sjogren s	+0.243	in primary	+0.206	patients with	+0.354	cirrhosis we	+0.285	of autoantibodies
+0.173	the same	+0.230	purified	+0.207	with	+0.120	is well	+0.207	however the	+0.242	autoantibodies to	+0.194	the presence	+0.327	mitochondrial	+0.282	occasionally
+0.169	of antibodies		mitochondrial		mitochondrial	+0.118	a variety	+0.204	reported to	+0.239	the mitochondrial	+0.179	are prominent		antigens		reported
+0.165	7 of	+0.226	the development	+0.196	immunodominant	+0.116	enzymes are	+0.192	is unknown	+0.230	concordant with	+0.176	have been	+0.266	showing	+0.239	the mitochondrial
+0.161	and branch	+0.206	issue we		epitopes	+0.111	specific nuclear		autoantibodies	+0.204	well known	+0.172	be found		reactivity	+0.235	of primary
+0.161	development of	+0.202	report that	+0.191	with a	+0.108	syndrome is	+0.174	were	+0.203	self antigens	+0.172	well known	+0.259	patients with	+0.216	mitochondrial
+0.160	autoantibodies to	+0.202	of 96	+0.186	mitochondrial 2	+0.102	were directed	+0.172	previously		the	+0.170	of antibodies	+0.255	and branch		antigens
+0.156	the	+0.202	7 positive	+0.163	a variety	+0.102	often been		characterized		autoantibodies	+0.150	reacting with	+0.212	of autoantibodies	+0.212	are prominent
	autoantibodies	+0.201	been occasionally	+0.157	reported although	+0.101	been occasionally	+0.170	that	+0.182	and or		mitochondrial	+0.210	we hypothesize	+0.210	at clinical
+0.149	for the	+0.194	of the	+0.153	antigens however	+0.095	the presence		abnormalities	+0.159	primary biliary	+0.149	antigens	+0.204	of the	+0.198	dehydrogenase
	with	+0.192	at clinical	+0.153	presence of	+0.094	clinical	+0.162	the presence		dehydrogenases	+0.146	known to		autoantibodies		and
+0.141	mitochondrial	+0.191	reacted with	+0.150	dehydrogenase		cirrhosis	+0.159	prominent in	+0.157	sjogren	+0.141	these latter	+0.196	were	+0.196	be found
+0.140	with sjogren	+0.190	target antigens		these	+0.091	presence		showing	+0.157	for the	+0.141	of primary		the	+0.195	96 patients
+0.132	those patients		antibodies	+0.150	known to	+0.089	autoantibodies to	+0.146	reactivity	+0.156	of 96		mitochondrial	+0.184	autoantibodies		previously
	antibodies	+0.188	directed		mitochondrial		including	+0.135	of 96	+0.153	known for	+0.133	target	+0.180	the development	+0.190	characterized
+0.130	directed		moreover	+0.149	antigens	+0.088	antimitochondrial		ketoacid	+0.141	we report		antibodies	+0.173	syndrome have		the
+0.128	syndrome the	+0.168	patients	+0.146	epitopes that	+0.086	cirrhosis we	+0.132	dehydrogenase	+0.141	patients showing	+0.132	directed	+0.167	prominent in	+0.189	autoantibodies
+0.127	7 positive	+0.165	patients with	+0.143	these latter	+0.085	been reported	+0.130	and or		mitochondrial	+0.132	and or	+0.157	significance is	+0.179	patients showing
+0.127	and primary	+0.160	including	+0.142	of other	+0.083	antigens the	+0.127	that the	+0.138	target	+0.125	self antigens	+0.144	reacting with	+0.176	development of
+0.126	reported although		antimitochondrial	+0.138	of both	+0.081	of primary	+0.126	cirrhosis to		moreover	+0.117	at specific		with	+0.175	reacted with
+0.121	syndrome and	+0.155	a group	+0.134	of antibodies	+0.079	7	+0.126	the mitochondrial	+0.132	patients	+0.116	and branch	+0.140	mitochondrial	+0.172	studied in
+0.120	branch chain	+0.154	patients reacted	+0.133	self antigens	+0.079	of 96	+0.121	of primary	+0.127	we studied	+0.113	of 96	+0.135	self antigens		dehydrogenase
+0.117	we report	+0.147	and to	+0.130	the development		with	+0.119	oxo acid	+0.127	2 oxo		ketoglutarate	+0.135	their significance	+0.154	these
+0.112	be concordant	+0.146	self antigens	+0.128	significance is	+0.076	mitochondrial		mitochondrial	+0.124	development of	+0.112	dehydrogenase		previously	+0.148	or that
	unknown	+0.145	acid	+0.126	to the	+0.074	well known	+0.117	target	+0.122	directed at	+0.110	syndrome have	+0.135	characterized	+0.144	and primary
+0.110	moreover		dehydrogenases	+0.122	of 96	+0.072	of autoantibodies	+0.113	patients reacted	+0.118	positive patients	+0.110	specific nuclear	+0.132	with a	+0.144	of 96
+0.106	self antigens	+0.144	syndrome have	+0.118	antigens the	+0.069	patients with	+0.112	target antigens	+0.116	showing		moreover		ketoglutarate		alpha
+0.105	reacted with	+0.144	are at	+0.117	chain ketoacid	+0.069	however the	+0.111	moreover in	+0.110	reactivity	+0.109	patients	+0.132	dehydrogenase	+0.142	ketoglutarate
+0.102	hypothesize that	+0.143	known to	+0.116	are prominent	+0.069	primary biliary	+0.106	one of	+0.109	with a	+0.106	to alpha	+0.129	2 oxo	+0.140	well known
+0.100	is well		dehydrogenase	+0.110	patients showing	+0.064	directed at		pyruvate	+0.109	reported although	+0.104	latter enzymes	+0.127	in those	+0.133	autoantibodies to
+0.099	with a	+0.140	and		antibodies	+0.064	branch chain	+0.104	dehydrogenase	+0.109	that the	+0.103	nuclear antigens	+0.127	of primary	+0.130	to purified
+0.098	antigens however	+0.131	to alpha	+0.107	directed		moreover	+0.101	been previously	+0.108	a variety	+0.091	with a	+0.120	both pyruvate	+0.128	both pyruvate
+0.098	are at	+0.126	moreover in	+0.104	patients with	+0.061	patients	+0.101	has often		occasionally	+0.091	presence of	+0.115	antibodies has	+0.128	of other
+0.096	been previously	+0.123	other self		the	+0.060	development of	+0.101	syndrome and	+0.102	reported	+0.089	has often		dehydrogenase	+0.122	cirrhosis to
+0.094	be found	+0.121	address this	+0.103	autoantibodies	+0.060	we hypothesize	+0.095	was known		dihydrolipoamide	+0.088	development of	+0.110	and	+0.120	a group
+0.093	remaining 6	+0.120	7 of	+0.102	be found	+0.059	that the	+0.093	reported although	+0.102	acetyltransferases	+0.086	to have	+0.100	would be	+0.115	reacting with
+0.093	directed at	+0.110	is unknown	+0.102	are at	+0.058	on liver	+0.091	concordant with	+0.102	have		alpha	+0.100	latter enzymes	+0.115	was known
+0.093	in those		antimitochondrial	+0.100	s syndrome	+0.057	with primary	+0.091	are prominent	+0.101	report that	+0.084	ketoglutarate		ketoacid		showing
+0.090	a variety	+0.110	antibodies	+0.095	patients are	+0.056	are the	+0.088	both pyruvate	+0.097	patients are		including	+0.100	dehydrogenase	+0.113	reactivity
+0.086	we studied	+0.109	concordant with	+0.092	has often		cirrhosis	+0.086	in a	+0.096	nuclear antigens	+0.082	antimitochondrial	+0.099	moreover	+0.108	however the
	including	+0.109	the same	+0.091	ono acid	+0.056	moreover		immunodominant	+0.093	target antigens	+0.082	cirrhosis to		patients	+0.106	been reported
+0.084	antimitochondrial	+0.107	the presence	+0.089	although their	+0.054	have primary	+0.083	epitopes	+0.092	antibodies has	+0.079	reported to	+0.099	one of	+0.099	to be

Figure 5.5 Explanations of OHSUMED data using ELI5.

5.2.2 Global Surrogate Models

We explained previously how some local interpretable methods are working using feature importance. Alternatively, we explored the use of global explainability using three interpretable models: Linear Regression (LR), Stochastic Gradient Descent (SGD) and Decision Tree (DT) based surrogate models. This method provides the general behavior or the big picture of the black-box model, i.e. working on the complete set of data. Besides, it is used to dedicate its linearity behavior, such as LR and SGD, to describe the black-box predictions globally. Accordingly, we used MimicExplainer modules that are implemented specifically for global surrogates in Python. Figure 5.6 Shows which kind of features that generally attributed during the classification decisions using the three surrogate models.

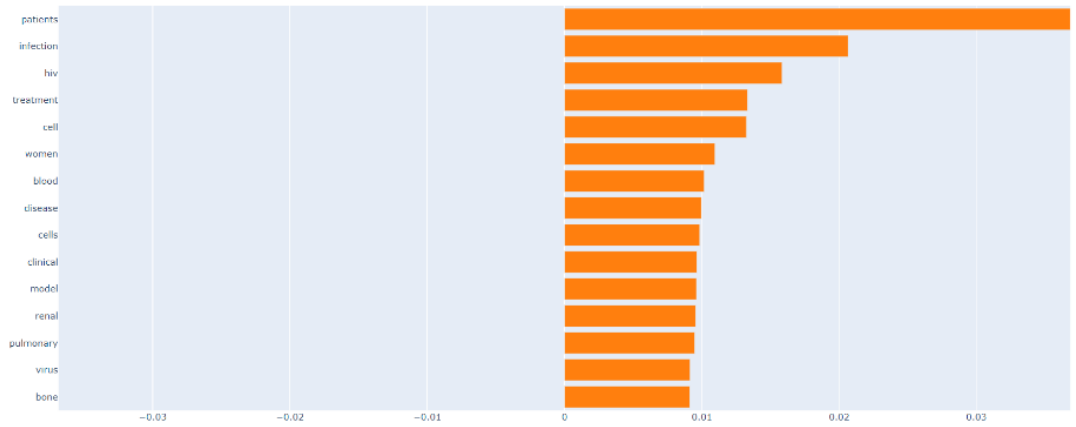
Figure 5.6 shows the global explanations taking into consideration the following observations:

1. The top feature for both SGD and LR is the word “patients” with an importance equal to 0.037 and 0.057 respectively. While DT consider the word “pituitary” as the most important feature in the whole dataset with a score equal to 1.269*1038.

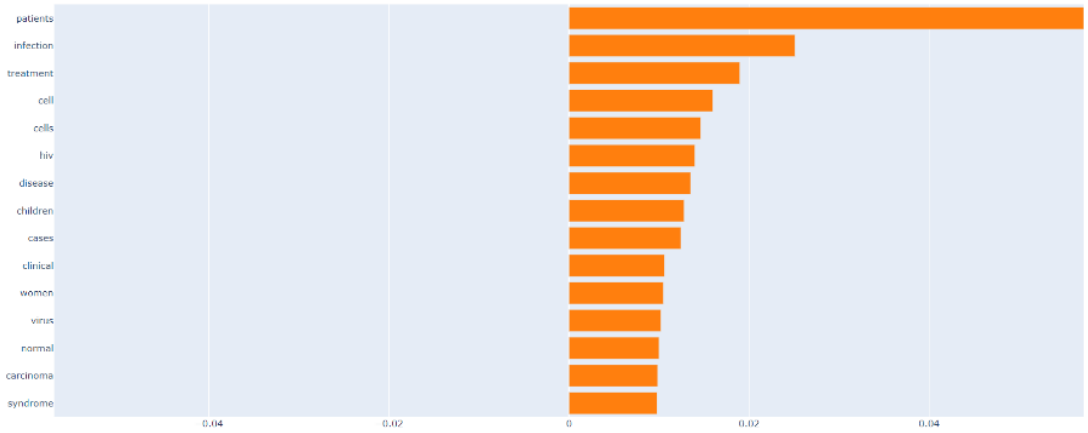
2. The features that are selected by the three surrogate models are distinguishable. That is, the nature of each surrogate model may play an important role to decide which feature is responsible for the predictions of the black-box model.

5.2.3 Model with Explainability Nature

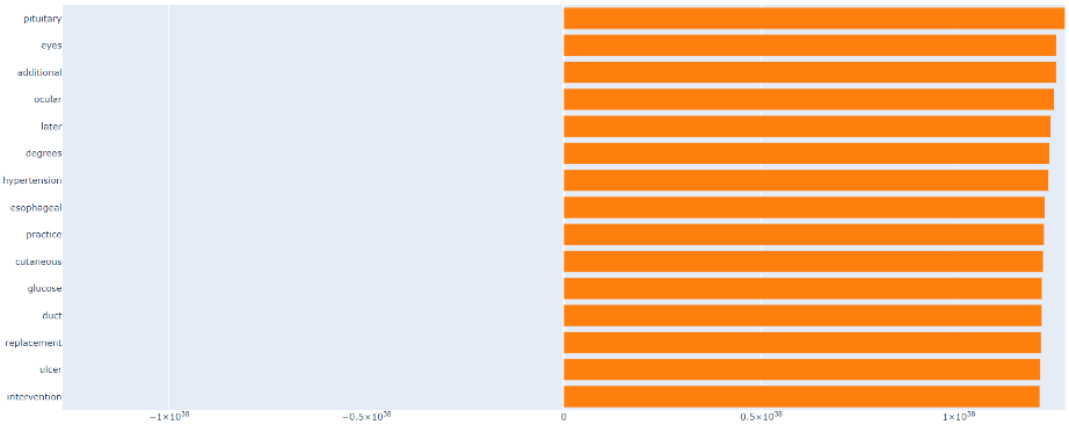
We decided also to see how global interpretable models can use their explainability nature on the OHSUMED dataset. To that end, we used DT to be our direct classifier regardless of its disappointing accuracy in order to illustrate its predictions’ decisions. Figure 5.7 Shows the visualization of DT classifier on OHSUMED dataset for an ML-Classification problem.



(a) SGD



(b) LR



(c) DT

Figure 5.6 Explanations of OHSUMED data using global surrogate models

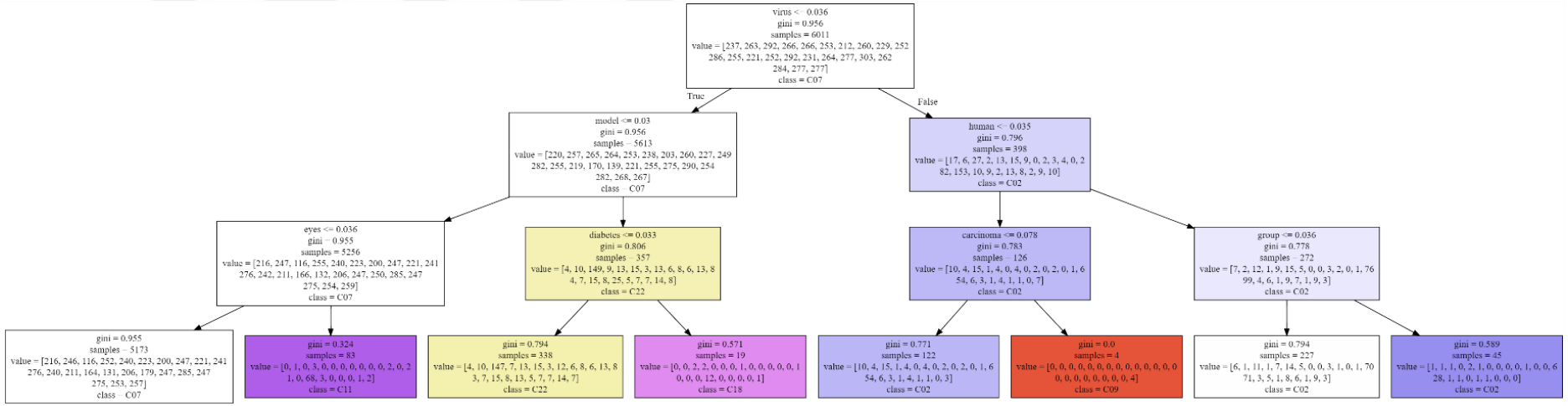


Figure 5.7 A DT as model with explainability nature.

Since our data is unstructured, DT shows all the decisions of the model in terms of the contributed features. Each node of the DT mirrors the feature in that node with two branches: TRUE or FASLE decisions, which decide a number of some contributed features. Moreover, it uses the Gini splitting method. Subsequently, the decisions and the features behind selecting class C07 are Virus, Model, and Eyes. Each has its own Gini score that helped to select that features in the decision process. Moreover, each class is described by a particular color. For example, class C07 has a white color, class C22 has a light brown color, class C02 has a light blue color, etc.

5.2.4 Discussions and Comparisons of the Results

As a general conclusion, each XAI has its own cons and pros. Table 5.3 shows all the advantages and disadvantages as we noticed during the work with XAI methods.

Table 5.3 A comparison between different XAI methods

Type	**XAI Method**	**Provided Info.**	**Class Inter-relationship**	**Feature-Disease Relevance**
Local	LIME	Predictions probabilities.	Strong	Strong
		Feature imp. scores. (+ve and –ve contribution).		
		Identifying features in the text.		
	SHAP	The mean of Shap value for each feature.	Strong	Medium
		Each class represented by a different color.		
		Distribution of features among classes.		
	ELI5	A table for all classes predictions probabilities.	Strong	Strong
		The mean of total importance scores.		
		Feature imp. scores. (+ve and –ve contribution).		
Global Surrogate		A table for all classes predictions probabilities.	None	Weak
		The mean of total importance scores.		
		Feature imp. scores. (+ve and –ve contribution).		
A Model With XAI Nature		The top features contributed globally as tree.	Weak	Weak
		Gini scores for each feature that identify the decision in the tree.		

5.3 Results and Discussion of Study 3

This section summarizes our findings in terms of two different factors: document and feature distribution-based parameters. As previously stated in Study 3 experiment section, the change in the number of documents can be used as an indicator of XAI complexity, while the change in the number of features can be used as a performance metric to indicate whether the classifier is performing well or poorly.

5.3.1 Document Scalability Effect on the Evaluation Metrics

We show the effect of increasing the number of documents, primarily from 100 to the entire 6900 sets of data. Figure 5.8 depicts how the DT's total depth, Average weighted class-based depth and Accuracy vary with document size. We discovered that the average weighted class-based depth, total depth, and accuracy are all proportional to the number of documents. For both SHAP and LIME, the three metrics scores increase and decrease as the number of documents increases or decreases, respectively. Furthermore, we found that SHAP outperforms LIME in terms of DT complexity (as measured by the average weighted class depth and total DT depth) and performance (as measured by DT accuracy). Furthermore, there is a significant difference in DT accuracy between SHAP and LIME However, both SAHP and LIME show a slight difference in terms of the average weighted class-based depth and DT total depth, although the former performed better.

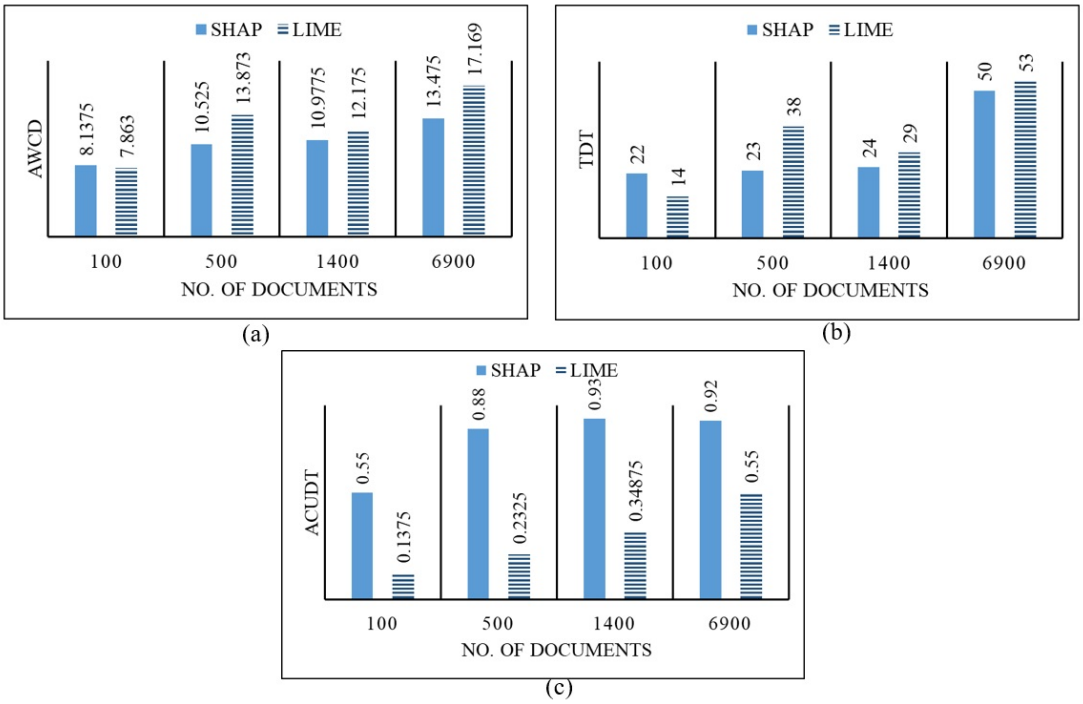


Figure 5.8 The results of the XAI measures with each set of documents.

5.3.2 Feature Distribution Effect on the Evaluation Metrics

Figure 5.9 illustrates how the number of features played an important role in the evaluation process. We conclude that both complexity metrics, average weighted class depth and DT total depth, produce the highest scores with SHAP. As a result, SHAP receives the lowest scores for both depth metrics and is thus less complex than LIME. The complexity of both XAI methods, on the other hand, is inversely proportional to the number of features. In other words, as the number of features decreases, the complexity of XA increases, and vice versa. On the other hand, increasing the number of features increases DT accuracy and vice versa. We also found that the differences in complexity metrics between SHAP and LIME is not significant, with only a three-level difference when 500 features are considered. While the DT accuracy differed noticeably between the two XAI methods, SHAP achieves the highest DT accuracy, 92% , using 500 and 200 features, whereas LIME achieves 55% and 54% using the same number of feature sets. Furthermore, as the number of top features increases, the tree transforms from a wide and sparse tree with

up to 500 features, to a deeper one with only 20 features.

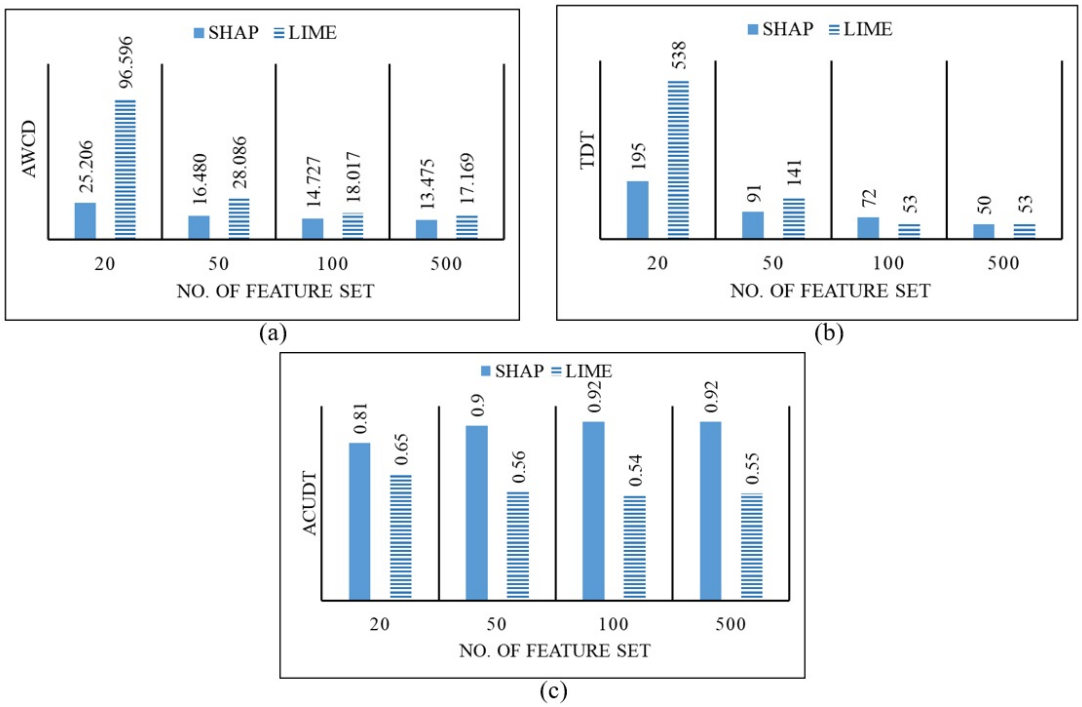


Figure 5.9 The results of the XAI measures with each set of features.

Generally, Figure 5.10 shows the big picture of the three evaluation metrics on SHAP and LIME. It illustrates that SHAP, in general, outperforms LIME in terms of the three measurements.

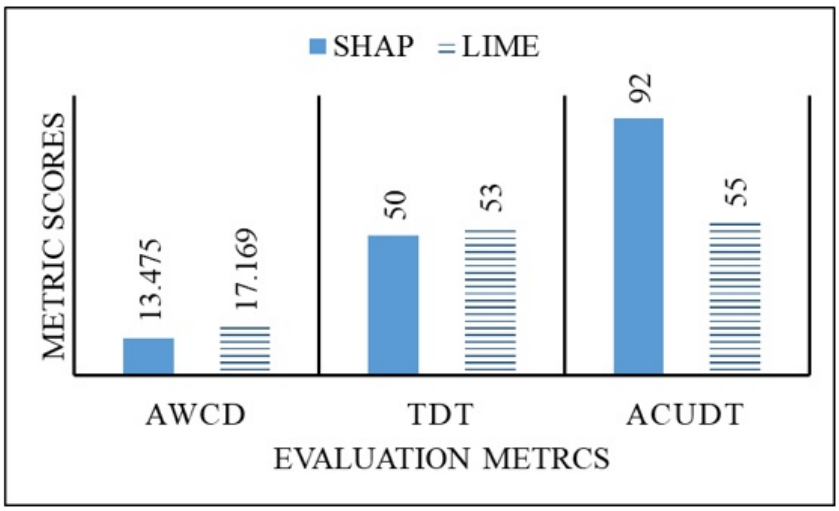


Figure 5.10 A comparison of SHAP and LIM for each evaluation metric.

5.3.3 Discussions and Comparisons of the Results

Based on the previous results, we concluded that SHAP outperforms LIME throughout all three evaluation metrics. The nature of calculating feature importance scores differs between the two XAI methods. These distinctions could explain the main differences between the two methods. Furthermore, as we discussed in section 2.3, most of the literature research potentially supports SHAP outperforming LIME in other quantitative and qualitative evaluation metrics (Messalas et al., 2020)(Antwarg et al., 2019)(Jesus et al., 2021).

Table 5.4 TDT and AWCD evaluation metrics on different sets of features.

No. of Top Features	**20**	**50**	**100**	**500**
TDT Metric	22	22	22	22
AWCD Metric	9.75	8.61	8.51	**8.14**

Moreover, the total depth of the decision tree and the average weighted class depth are good indicators of the complexity of the XAI method. Figure 5.10 shows that SHAP has a more stable and lower total tree depth than LIME. However, the AWCD metric is more specific than TDT in determining the degree of complexity. Table 5.4 shows, for example, how the TDT metric is stable and does not change; TDT equals 22 levels deep because different sets of features were used. While TDT is stable, AWCD can show differences in class depths based on each leaf depth, because the depth of each class can be found at a different level, and hence the average of the depth's weighted sum can be more useful in defining complexity measurement.

Furthermore, document scalability is an indicator that reflects how XAI methods work with different numbers of documents. SHAP is more scalable and can handle documents of more varied sizes than LIME. On the other hand, feature distribution can help us learn more about the effects of important scores from SHAP and LIME on the decision tree's classification accuracy. This factor for different feature sets can demonstrate how SHAP, with its importance scores, is more efficient than LIME in positively affecting the decision tree's accuracy.

Figure 5.10 shows that the accuracy of the DT with the SHAP method is 92% on the full set of documents and 500 feature sizes, while LIME is at 55% . We also believe that this accuracy is an XAI accuracy and that it cannot be compared to the classification accuracy of the black-box model, i.e. the FNN network. Furthermore, XAI importance scores are generated based on how a specific black-box model responds to the entered data. Accordingly, we cannot claim that the decision tree is more robust and powerful than FNN in classifying OHSUMED documents because both accuracies serve different purposes and need different model configurations. Alternatively, the decision tree's XAI accuracy, i.e. the ACUDT metric, can be utilized as a guide to help us choose only the most significant features as an input to the black-box model (Marcilio & Eler, 2020).

For comparison, Table 5.5 provides a comparison between our XAI evalaution method, i.e. DT depth method, and other some recent papers in literature. Although there are some differences between our method and theirs, such as the use of different datasets and black-box models, yet they are comparing among several XAI methods as we do. The main aim of this comparison is to show how the other recent research drew the same conclusion as we did, which is that SHAP outperforms other XAI methods. Although some papers such as (Ramon et al., 2020) indicate that SEDC XAI method is the superior XAI method, they also proved in their research that SHAP was better than LIME as we did in our method. That is, on comparison, the effectiveness of SHAP tool is validated by not only the XAI evaluation metrics implemented in literature but also by our evaluation method.

Table 5.5 A comparison of our method and other XAI evaluation systems

Paper	**Dataset Type**	**XAI Methods**	**Evaluation Methods**	**The Best XAI Method**
Our Method	Text	LIME and SHAP	DT Depth	SHAP
Messalas et al. (2020)	Text	LIME and Model-Agnostic Shapley value explanations (MASHAP)	Runtime, Consistency Guarantees, Identity, Expressive Power, Translucency, and Portability	MASHAP
Ramon et al. (2020)	Text	Counterfactual LIME-C, Counterfactual SHAP-C and Search for Evidence Counterfactuals (SEDC)	Effectiveness and Efficiency	SEDC
Antwarg et al. (2019)	Text	LIME and SHAP	Mean Reciprocal Rank (MRR) Measure	SHAP
ElShawi et al. (2020)	Tabular and Text	LIME, SHAP, Anchors, LORE, ILIME and MAPLE	Time, Trust,Identity, Stability, and Separability	Tabular: SHAP and Text: MAPLE
Jesus et al. (2021)	Text	LIME and SHAP	Application-Grounded	SHAP

Finally, researchers may debate about whether XAI representation, such as XAI visualization, should be included in the XAI evaluation process. However, this can sway human judgment and bias it toward a particular XAI method. Nevertheless, in our opinion, the most important part of the explanations are the features and their role in driving a specific prediction decision. As a result, the key issue behind any black-box explanation is how the XAI methods can define the most important features. According to (Jesus et al., 2021), "another important consideration is the biases that may emerge if explanation methods are distinguishable due to some factor (e.g., their representation). Preventing their representational differences is thus a necessary precaution toward isolating the quality and relevance of the explanation methods from all other potential visual factors". Hence, using quantitative XAI evaluation measures will be more robust in providing an interpretability assessment that is not influenced by human sentiments.

5.4 Results and Discussion of Study 4

When it comes to classifying the entire set of OHSUMED data, our FNN model achieves only 49.8% accuracy. It means that, our black-box model is less effective when all features are taken into account. However, for the sake of XAI, Study 3 was concentrating on evaluating XAI methods and comparing them in terms of quantitative evaluation measurements. As a result, the goal Study 3 is not to evaluate the performance of the black-box classifier, but rather to deal directly with the XAI methods and their complexity.

On the other hand, as it is elaborated in the experiment of Study 4, we tried to enhance the accuracy of FNN multi-class classification by employing SHAP as a dimensionality reduction mechanism. Accordingly, we got the inspiration from the ACUDT XAI metric to increase the efficiency of our black-box model. Therefore, the black-box accuracy is reaching its best, i.e. recording 92.1% mean accuracy with 1.96 standard deviation, when the number of features are reduced to 200 and the number of epochs is increased to 30. Table 5.6 shows all the results of using different number of

feature set with increasing the number of FNN epochs from 15 to 30.

Table 5.6 The accuracy after using SHAP as feature selection

No. of features	Accuracy score 10-fold cross validation					
	15 epochs		20 epochs		30 epochs	
	Mean %	Std. Deviation	Mean %	Std. Deviation	Mean %	Std. Deviation
100	62.42	5.69	79.71	6.97	91.40	1.99
200	64.83	6.50	79.56	2.80	**92.10**	1.96
300	56.36	9.67	77.01	5.16	91.69	1.31
400	56.01	6.04	76.04	5.07	91.12	2.59

According to the result section, the multi-class classification accuracy has been remarkably improved. The reason for this improvement is as follows:

Using a simple neural network architecture: A modest or complicated artificial neural network structure can be created depending on the problem and the data of the study (Szumega et al., 2021). In our case, the model's performance and efficiency is improved by simplifying the neural network structure. As a result, we converted our LSTM neural network, that we used in our study (Ahmed et al., 2020), to a basic feed forward structure FNN. We believe that as the structure becomes more complex, the weights and parameters become more complex as well, resulting in fluctuations in accuracy and error rate during the classification process.

Increasing the number of the neural network epochs: The number of epochs can sometimes, but not always, improve the neural network's performance (Ali et al. 2021). In our last study, we tried to increase the number of epochs but with using the whole set of features, relevant and irrelevant feature set, and a complicated neural network structure, LSTM neural network. We sought to expand the number of epochs in our previous study,(Ahmed et al., 2020), by employing the entire collection of data features, both relevant and irrelevant, as well as the use of a difficult neural network structure, the LSTM neural network. As a result, the accuracy remained constant at

50% and could not improve. In this study, increasing the epochs improved the neural network's performance, especially when the feature dimensionality was reduced and a simple neural network structure was applied.

The use of SHAP XAI as a dimensionality reduction: As we mentioned in the result section, using SHAP as a feature selection technique enhanced the performance of the multiclass classification model. It selects the most important and relevant features for each document in the OHSUMED dataset. Accordingly, the resulted features forced the classification model to be more discriminative so that it can identify the pattern of each class more appropriately (Abrol et al., 2021).

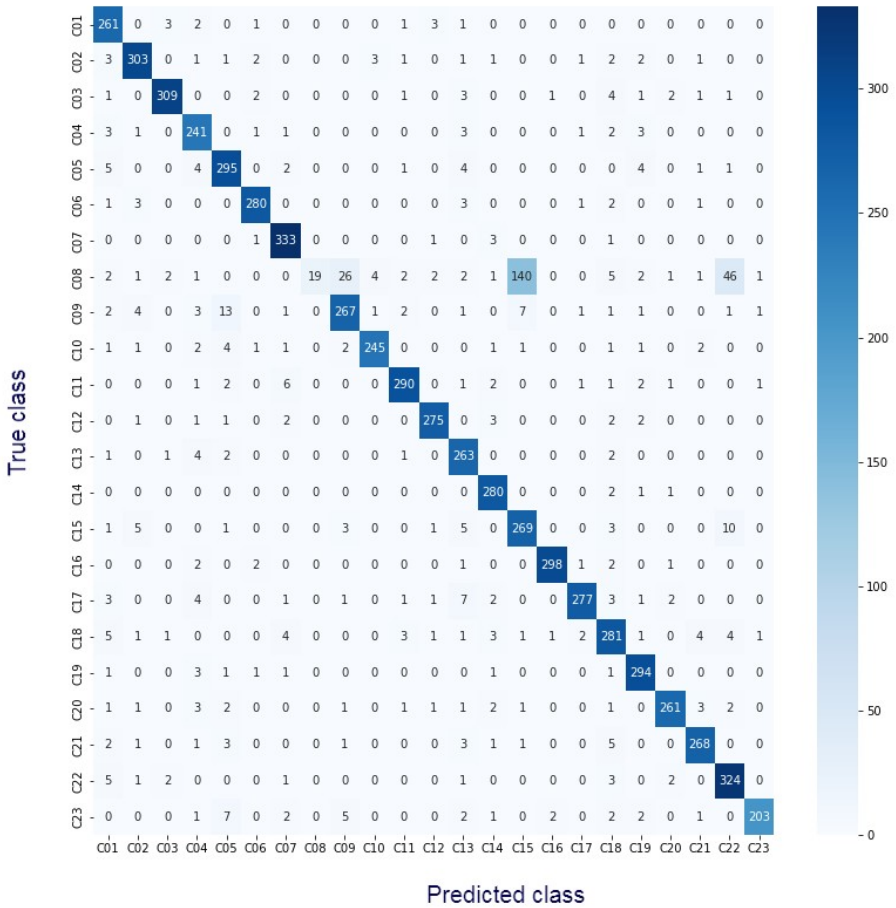


Figure 5.11 The confusion matrix of the best result.

Additionally, Figure 5.11 shows the confusion matrix related to the best score in Table 5.6, i.e. the 92.1%. The confusion matrix demonstrates that by employing dimensionality reduction for a given document, our model can classify the given features more effectively and precisely. However, the model identified 140 documents related to class C08 as C15 which implies that both classes are interrelated and have common features. On the other hand, class C08 has some common features with C22 with almost 46 documents.

In overall, we find that the multi-class classification model is being improved as the preceding three parameters are taken into account. Figure 5.12 shows the progress in our system accuracy during the work in this thesis.

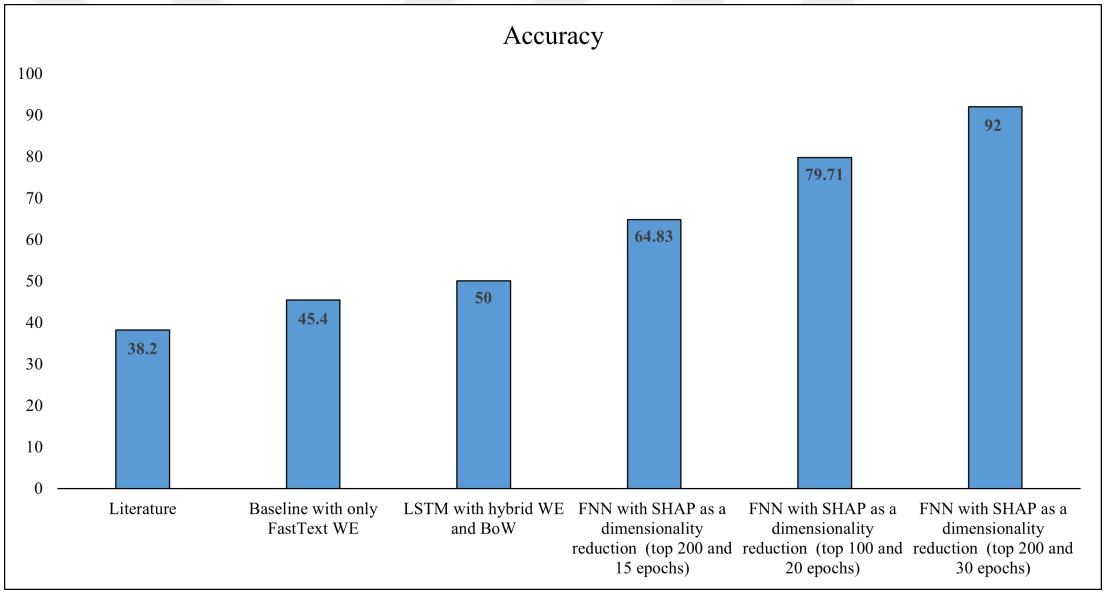


Figure 5.12 The progress of the XAI multi-class system results.

CHAPTER SIX
CONCLUSION

6.1 General Conclusion

With the existence of big data and current enhancements in deep learning, the performance of artificial intelligence (AI) is reaching and even exceeding the human levels by solving many complicated tasks. However, because of the non-linear structure of some successful machine learning and artificial intelligence models, these models are usually applied in a black box manner, i.e., no information is provided about what exactly makes them arrive at their predictions. Hence, the progress of developing methods for visualizing, explaining and interpreting deep learning models has recently attracted increasing attention. On the other hand, solving multi-class text classification problem becomes a must specially in biomedical fields. Therefore, this dissertation discusses several issues related with Explainable AI on a multi-class data problem.

The first section of this dissertation seeks to discover distinct features and patterns within the OHSUMED multi-class dataset, which is a collection of biological text files related to cardiovascular disorders. This might be achieved by upgrading the feature representation approach employed in the classification process' initial phase. Therefore, we developed a weighted feature representation method that combines the benefits of both techniques, i.e., word embeddings (WE) and bag-of-words (BoW). We derived WE FastText vectors from the PubMed and MIMIC III datasets.

Then, using the OHSUMED dataset, we created three primary statistical weighting scores for each word in the documents: term frequency (TF), inverse document frequency (IDF), and class probability (CP). Moreover, we evaluated our weighted vector feature on a case study modeled as a multi-class problem and employed it to predict one of the 23 cardiovascular disease categories. Our classification model, which is based on bidirectional LSTM and CP weighted feature-representation approach, achieved the greatest performance score. In addition, its outcomes were

superior to those of machine learning (ML) techniques and one related research in literature. It appears that class-term relationship information enhances the performance of a multiclass categorization system.

We noticed that our classification model performed better after adding essential information and weights to the initial WE vector. Furthermore, we discovered that the overall performance of categorizing the OHSUMED dataset was poor due to the fact that it describes various interrelated and associated heart and blood vessel diseases and conditions. In addition, the OHSUMED dataset has 23 classes, which complicates the categorization procedure. Therefore, the performance declines due to the enormous number of classes.

Additionally, the main adjective in the second part of this thesis is to explain the outcomes of multi-class classification model on the OHSUMED medical dataset. After exploring several familiar interpretable methods in the literature, we can conclude that local explainable techniques are more specific to identify the medical features more than the global ones. The features of the local interpretable methods, especially LIME and SHAP, are greatly contributed to the predictions. Besides, any physician or doctor will be more interested in seeing what the main medical entities and features in clinical text are, such as diagnosis and symptoms, that were behind a particular disease prediction. Another doctor will be also more interested in identifying the interrelationship between disease' classes and the features behind this dependency. Local explainable methods proved their effectiveness of discovering class dependency more than the global methods which give the total impact of certain features in the prediction process.

Moreover, some XAI strategies focused on feature importance scores to assign significance to features based on how responsive the black-box model is to that feature. As a result, some local and global XAI techniques rely entirely on that mechanism. Then, evaluating XAI methods becomes necessary, which can be done using either subjective or objective methods. Subjective or qualitative evaluation methods rely on human judgment and can be time-consuming and labor-intensive.

While objective or quantitative methods automatically evaluate explainability to save time and effort. Accordingly, our third objective in this thesis is to propose a quantitative XAI evaluation technique for measuring the complexity of the local-based XAI method, which is heavily reliant on feature importance scores. The depth of the decision tree (DT) is used as a complexity metric to assess and compare the complexities of two XAI methods: SHAP and LIME. The more complex the tree, the deeper it is. The explanations, or importance scores, of the XAI method, are fed into the decision tree. Following that, the tree will draw decisions based on the input explanations, allowing us to calculate its depth to measure complexity. Consequently, we developed two complexity metrics: one based on the total depth of the DT (TDT), and the other on the average weighted depth of each DT's leaf or class (AWCD). On the other hand, there is another metric that is based on the DT's accuracy (ACUDT) which can be used to determine the appropriate number of features to increase the performance of the DT classifier. The DT accuracy can help us figure out how to control the black-box accuracy's efficiency in the future. The results show that the DT of the SHAP XAI method is less deep and less complex than that of the LIME method. Furthermore, SHAP outperforms LIME in terms of document scalability and feature sensitivity. To ensure that we are on the right track, our findings agree with several studies that demonstrate SHAP's effectiveness, particularly when compared to LIME. We intend to compare some global XAI methods using our quantitative metric. Finally, we propose a dimensionality reduction depending on the feature importance scores of SHAP explainability XAI method. The results show that SHAP XAI improves the accuracy of the multiclass classification from 50% to approximately 92%.

6.2 Future Work

XAI is one of the most popular and recent topics that can open your imagination to solve certain challenge or problem. We intend first to enhance our feature representation method semantically. Therefore, we would like to use BERT

(Bidirectional Encoder Representations from Transformers) as one rich resource of information that can be accompanied with word embedding or Bag of Word feature representation systems to expand the medical knowledge semantically. Additionally, and for the sake of enhancing XAI, we intend to explain the inner behavior of the black-box model. This means that instead of describing how a black-box model behaves in response to input or output, we need to comprehend the model from the inside out. Another contribution can be taken into account, explaining the word embeddings vectors for the models that use WE systems as a primary step. This contribution can add much sense of using a particular WE system in any classification model.

REFERENCES

Abrol, A., Fu, Z., Salman, M., Silva, R., Du, Y., Plis, S., & Calhoun, V. (2021). Deep learning encodes robust discriminative neuroimaging representations to outperform standard machine learning. *Nature Communications*, *12*(1), 1–17.

Ahmed, N., Dilmaç, F., & Alpkocak, A. (2020). Classification of biomedical texts for cardiovascular diseases with deep neural network using a weighted feature representation method. In *Healthcare*, vol. 8, *Multidisciplinary Digital Publishing Institute*, 392.

Ahmed, N., Yigit, A., Isik, Z., & Alpkocak, A. (2019). Identification of leukemia subtypes from microscopic images using convolutional neural network. *Diagnostics*, *9*(3), 104.

Angelov, P., & Soares, E. (2020). Explainable-by-design approach for covid-19 classification via ct-scan. *MedRxiv*.

Antwarg, L., Shapira, B., & Rokach, L. (2019). Explaining Anomalies Detected by Autoencoders Using SHAP. *ArXiv*, 1–37.

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., et al. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, *58*, 82–115.

Blanco, A., Perez-de Viñaspre, O., Pérez, A., & Casillas, A. (2020). Boosting icd multi-label classification of health records with contextual embeddings and label-granularity. *Computer Methods and Programs in Biomedicine*, *188*, 105264.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146.

Carvalho, D. V., Pereira, E. M., & Cardoso, J. S. (2019). Machine learning interpretability: A survey on methods and metrics. *Electronics*, *8*(8), 832.

Chen, Q., Peng, Y., & Lu, Z. (2019). Biosentvec: creating sentence embeddings for biomedical texts. In *2019 IEEE International Conference on Healthcare Informatics (ICHI)*, *IEEE*, 1–5.

Doshi-Velez, F., & Kim, B. (2018). Considerations for evaluation and generalization in interpretable machine learning. In *Explainable and Interpretable Models in Computer Vision and Machine Learning*, 3–17.

Dunn, J., Mingardi, L., & Zhuo, Y. D. (2021). Comparing interpretability and explainability for feature selection. *ArXiv Preprint ArXiv:2105.05328*.

ElShawi, R., Sherif, Y., Al-Mallah, M., & Sakr, S. (2020). Interpretability in healthcare: A comparative study of local machine learning interpretability techniques. *Computational Intelligence*, 1–18.

Enríquez, F., Troyano, J. A., & López-Solaz, T. (2016). An approach to the use of word embeddings in an opinion classification task. *Expert Systems with Applications*, *66*, 1–6.

Feng, J., Shaib, C., & Rudzicz, F. (2020). Explainable Clinical Decision Support from Text. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1478–1489.

Gargiulo, F., Silvestri, S., Ciampi, M., & De Pietro, G. (2019). Deep neural network for hierarchical extreme multi-label text classification. *Applied Soft Computing*, *79*, 125–138.

Ge, W., Huh, J.-W., Park, Y. R., Lee, J.-H., Kim, Y.-H., & Turchin, A. (2018). An Interpretable ICU Mortality Prediction Model Based on Logistic Regression and Recurrent Neural Networks with LSTM units. In *AMIA ... Annual Symposium proceedings. AMIA Symposium*, vol. 2018, 460–469.

Ghassemi, M., Oakden-Rayner, L., & Beam, A. L. (2021). The false hope of current approaches to explainable artificial intelligence in health care. *The Lancet Digital Health*, *3*(11), e745–e750.

Godin, F., Demuynck, K., Dambre, J., De Neve, W., & Demeester, T. (2018). Explaining character-aware neural networks forword-level prediction: Do they discover linguistic rules? *ArXiv*, 3275–3284.

Hailemariam, Y., Yazdinejad, A., Parizi, R. M., Srivastava, G., & Dehghantanha, A. (2020). An Empirical Evaluation of AI Deep Explainable Tools. *2020 IEEE Globecom Workshops, GC Wkshps 2020 - Proceedings*, *2*, 0–5.

He, T., Hu, J., Song, Y., Guo, J., & Yi, Z. (2020). Multi-task learning for the segmentation of organs at risk with label dependence. *Medical Image Analysis*, *61*, 101666.

Hearty, Á. P., & Gibney, M. J. (2008). Analysis of meal patterns with the use of supervised data mining techniques - Artificial neural networks and decision trees. *American Journal of Clinical Nutrition*, *88*(6), 1632–1642.

Hu, D., Chen, M., Wang, T., Chang, J., Yin, G., Yu, Y., & Zhang, Y. (2018). Recommending similar bug reports: a novel approach using document embedding model. In *2018 25th Asia-Pacific Software Engineering Conference (APSEC)*, *IEEE*, 725–726.

Islam, S. R., Eberle, W., & Ghafoor, S. K. (2020). Towards quantification of explainability in explainable artificial intelligence methods. In *The Thirty-Third International Flairs Conference*.

Jesus, S., Belém, C., Balayan, V., Bento, J., Saleiro, P., Bizarro, P., & Gama, J. (2021). How can i choose an explainer?: An Application-grounded Evaluation of Post-hoc Explanations. *FAccT 2021 - Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 805–815.

Jha, K., Wang, Y., Xun, G., & Zhang, A. (2018). Interpretable word embeddings for medical domain. In *2018 IEEE International Conference on Data Mining (ICDM)*, *IEEE*, 1061–1066.

Johnson, A. E., Pollard, T. J., Shen, L., Li-Wei, H. L., Feng, M., Ghassemi, M., Moody,

B., Szolovits, P., Celi, L. A., & Mark, R. G. (2016). Mimic-iii, a freely accessible critical care database. *Scientific Data*, *3*(1), 1–9.

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information*, *10*(4), 150.

Kumar, B. N., MSVSB, R., & Vardhan, B. V. (2018). Enhancing the performance of an intrusion detection system through multi-linear dimensionality reduction and multi-class svm. *International Journal of Intelligent Engineering and Systems*, *11*(1), 181–192.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *International Conference on Machine Learning*, *PMLR*, 1188–1196.

Lei, Y., Dogan, Ü., Zhou, D.-X., & Kloft, M. (2019). Data-dependent generalization bounds for multi-class classification. *IEEE Transactions on Information Theory*, *65*(5), 2995–3021.

Li, C., Zhang, M., Bendersky, M., Deng, H., Metzler, D., & Najork, M. (2019). Multi-view embedding-based synonyms for email search. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 575–584.

Lim, T. S., Tay, K. G., Huong, A., & Lim, X. Y. (2021). Breast cancer diagnosis system using hybrid support vector machine-artificial neural network. *International Journal of Electrical & Computer Engineering (2088-8708)*, *11*(4).

Lin, Z. Q., Shafiee, M. J., Bochkarev, S., Jules, M. S., Wang, X. Y., & Wong, A. (2019). Do explanations reflect decisions? a machine-centric strategy to quantify the performance of explainability algorithms. *ArXiv Preprint ArXiv:1910.07387*.

Liu, C.-z., Sheng, Y.-x., Wei, Z.-q., & Yang, Y.-Q. (2018). Research of text classification based on improved tf-idf algorithm. In *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)*, *IEEE*, 218–222.

Luckey, D., Fritz, H., Legatiuk, D., Peralta Abadía, J. J., Walther, C., & Smarsly, K. (2022). Explainable artificial intelligence to advance structural health monitoring. In *Structural Health Monitoring Based on Data Science Techniques*, 331–346.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 4768–4777.

Marcílio, W. E., & Eler, D. M. (2020). From explanations to feature selection: assessing shap values as feature selection mechanism. In *2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, *IEEE*, 340–347.

Marcilio, W. E., & Eler, D. M. (2020). From explanations to feature selection: Assessing SHAP values as feature selection mechanism. *Proceedings - 2020 33rd SIBGRAPI Conference on Graphics, Patterns and Images, SIBGRAPI 2020*, 340–347.

Melamud, O., Goldberger, J., & Dagan, I. (2016). context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning*, 51–61.

Messalas, A., Aridas, C., & Kanellopoulos, Y. (2020). Evaluating MASHAP as a faster alternative to LIME for model-agnostic machine learning interpretability. In *Proceedings - 2020 IEEE International Conference on Big Data, Big Data 2020*, 5777–5779.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *ArXiv Preprint ArXiv:1301.3781*.

Mohseni, S., Block, J. E., & Ragan, E. (2021). Quantitative evaluation of machine learning explanations: A human-grounded benchmark. In *26th International Conference on Intelligent User Interfaces*, 22–31.

Molnar, C. (2019). *Interpretable Machine Learning*. https://christophm.github.io/interpretable-ml-book/.

Monica, K. M., & Parvathi, R. (2019). Traditional dimensionality reduction techniques using deep learning. *International Journal of Recent Technology and Engineering*, *8*(3), 7153–7160.

Pagliardini, M., Gupta, P., & Jaggi, M. (2017). Unsupervised learning of sentence embeddings using compositional n-gram features. *ArXiv Preprint ArXiv:1703.02507*.

Panigrahi, A., Simhadri, H. V., & Bhattacharyya, C. (2019). Word2sense: sparse interpretable word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 5692–5705.

Pavlov, Y. L. (2019). *Random forests*.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Ramon, Y., Martens, D., Provost, F., & Evgeniou, T. (2020). A comparison of instance-level counterfactual explanation algorithms for behavioral and textual data: SEDC, LIME-C and SHAP-C. *Advances in Data Analysis and Classification*, *14*(4), 801–819.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Model-agnostic interpretability of machine learning. *ArXiv Preprint ArXiv:1606.05386*.

Schmidt, C. W. (2019). Improving a tf-idf weighted document vector embedding. *ArXiv Preprint ArXiv:1902.09875*.

Şenel, L. K., Utlu, I., Yücesoy, V., Koc, A., & Cukur, T. (2018). Semantic structure and interpretability of word embeddings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *26*(10), 1769–1779.

Shen, Y., Zhang, Q., Zhang, J., Huang, J., Lu, Y., & Lei, K. (2018). Improving medical short text classification with semantic expansion using word-cluster embedding.

In *International Conference on Information Science and Applications*, *Springer*, 401–411.

Singh, A., Sengupta, S., & Lakshminarayanan, V. (2020). Explainable deep learning models in medical image analysis. *Journal of Imaging*, *6*(6), 52.

Singh, D., Kumar, V., Yadav, V., & Kaur, M. (2021). Deep neural network-based screening model for covid-19-infected patients using chest x-ray images. *International Journal of Pattern Recognition and Artificial Intelligence*, *35*(03), 2151004.

Sinoara, R. A., Camacho-Collados, J., Rossi, R. G., Navigli, R., & Rezende, S. O. (2019). Knowledge-enhanced document embeddings for text classification. *Knowledge-Based Systems*, *163*, 955–971.

Soares, E. (2020). Explainable-by-design approach for covid-19 classification via ct-scan. *MedRxiv*, *41*.

Soliman, A. B., Eissa, K., & El-Beltagy, S. R. (2017). Aravec: A set of arabic word embedding models for use in arabic nlp. *Procedia Computer Science*, *117*, 256–265.

Sunagar, P., Kanavalli, A., & Shetty, N. D. (2020). Feature extraction and selection techniques for text classification: A survey. *International Journal of Advanced Research in Engineering and Technology IJARET*, *11*, 2871–2881.

Szumega, J. M., Boukabache, H., & Perrin, D. (2021). Neural network approach for efficient calculation of the current correction value in the femtoampere range for a new generation of ionizing radiation monitors at cern. *Radiation Physics and Chemistry*, *188*, 109539.

Tan, S. (2007). Large margin drag pushing strategy for centroid text categorization. *Expert Systems with Applications*, *33*(1), 215–220.

Tehrani, A. F., & Ahrens, D. (2017). Modeling label dependence for multi-label classification using the choquistic regression. *Pattern Recognition Letters*, *92*, 75–80.

Teng, F., Yang, W., Chen, L., Huang, L. F., & Xu, Q. (2020). Explainable Prediction of Medical Codes With Knowledge Graphs. *Frontiers in Bioengineering and Biotechnology*, *8*(August), 1–11.

Van Der Waa, J., Nieuwburg, E., Cremers, A., & Neerincx, M. (2021). Evaluating xai: A comparison of rule-based and example-based explanations. *Artificial Intelligence*, *291*, 103404.

Wang, Y., Liu, S., Afzal, N., Rastegar-Mojarad, M., Wang, L., Shen, F., Kingsbury, P., & Liu, H. (2018). A comparison of word embeddings for the biomedical natural language processing. *Journal of Biomedical Informatics*, *87*, 12–20.

William Hersh and colleagues at the Oregon Health Science University (1998). *Ohsumed*. Retrieved March 5, 2020, from http://disi.unitn.it/moschitti/corpora/ohsumed-first-20000-docs.tar.gz.

Ye, X., Shen, H., Ma, X., Bunescu, R., & Liu, C. (2016). From word embeddings to document similarities for improved information retrieval in software engineering. In *Proceedings of the 38th International Conference on Software Engineering*, 404–415.

Ye, Z., Li, F., & Baldwin, T. (2018). Encoding sentiment information into word vectors for sentiment analysis. In *Proceedings of the 27th International Conference on Computational Linguistics*, 997–1007. from https://aclanthology.org/C18-1085.

Zhang, J.-J., Fang, M., Wang, H., & Li, X. (2015). Dependence maximization based label space dimension reduction for multi-label classification. *Engineering Applications of Artificial Intelligence*, *45*, 453–463.

Zhang, Y., Henao, R., Gan, Z., Li, Y., & Carin, L. (2018). Multi-label learning from medical plain text with convolutional residual models. In *Machine Learning for Healthcare Conference*, *PMLR*, 280–294.

Zhou, J., Gandomi, A. H., Chen, F., & Holzinger, A. (2021). Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics (Switzerland)*, *10*(5), 1–19.

Zhou, W., Wang, H., Sun, H., & Sun, T. (2019). A method of short text representation based on the feature probability embedded vector. *Sensors*, *19*(17), 3728.