Advancing Medical Diagnosis: A Comparative Study of Machine Learning Models and Text Embedding Techniques

Arthur Chansel, Marianne Scoglio & Gilles de Waha
Class Project 2, CS-433 Machine Learning, EPFL

***Abstract*— Recently, Large Language Models have gained attention as a potentially superior solution for medical diagnosis using machine learning, especially in environments with limited healthcare resources. Our study explored the efficacy of LLMs by comparing them against various traditional machine learning models and text embedding techniques, focusing on their capability to interpret medical symptoms in data-sparse settings. Our results indicated that both traditional models and neural networks tended to overfit and did not achieve satisfactory F1 score results, a challenge exacerbated by the limited size of our dataset. These findings highlight the potential of LLMs like MEDITRON to enhance performance and robustness in medical diagnostics. Additionally, our approach addressed ethical and privacy considerations related to machine learning in healthcare.**

I. Introduction

In developing countries, the challenges of limited healthcare resources and access to medical expertise significantly hinder effective medical diagnosis. The deployment of machine learning models as a solution to these challenges has become increasingly popular. Recently, Large Language Models (LLMs) have emerged as a potentially superior solution for medical diagnosis in these settings. With our work, we aim to assess the efficiency of LLMs by conducting comparative analyses with different established machine learning models and text embedding approaches. Our reference LLM, MEDITRON[1], builds on Llama-2 and extends pretraining on internationally-recognized medical guidelines in particular. The dataset we used generated patients from these guidelines. For each disease, we had three sets of associated symptoms. We sometimes ended up with less or more than three patients, for example when a disease was mentioned in several guidelines. With this dataset, our goal was to discover how accurate we could be in the medical diagnosis without the use of LLMs. We experimented with a range of classical models, including Random Forest, Linear Support Vector Classifier, and Gaussian Naive Bayes, as well as neural networks. Additionally, we employed various text transformations and embeddings, such as one-hot encoding, TF-IDF, word2vec, doc2vec, and BERT.

II. Models and methods

A. Data Preprocessing

First, we handled the data type in order to transform the data structure into a suitable format for further methods. Then, we had to deal with missing values that occur in the data set. They are of several kind : missing condition name, missing list of symptoms, list with no symptoms. We considered them all as outliers and removed them.

B. Data Representation

The features as list of symptoms are not directly usable for ML models. We first had to represent them in a suitable format. To understand better our data set, here is an example of the three sets of symptoms for the condition "Lung Cancer":

patients	symptoms			
1	Persistent cough	Hemoptysis	Localized chest pain	Unexplained weight loss
2	Chronic cough	Dull ache		
3	Persistent cough	Low-grade fever	Night sweats	Mild chest discomfort

1) OHE: One Hot Encoding is our baseline approach. We considered unique symptoms name for binary encoding. This is a classic method, which is very fast to compute but generates a high dimension data set.
However, as demonstrated on the example above, there are multiple ways to describe the same symptoms (persistent cough and chronic cough, for example), so it is important to detect similar symptoms (localized chest pain and mild chest discomfort, for example) and achieve semantic proximity for good prediction. For these 2 reasons, one-hot encoding is not the best data representation.

For the following data representations, we combined for every patient their symptoms into one string.

2) TF-IDF: Term Frequency-Inverse Document Frequency is a statistical measure used to evaluate how significant a word is to a patient's symptoms within the list of all the symptoms. It combines Term Frequency (TF), indicating how often a word appears in a patient's symptoms, with Inverse Document Frequency (IDF), which assesses the word's rarity across all symptoms. The TF-IDF score, a product of TF and IDF, reflects the word's relevance, giving higher importance to rare yet significant terms and less to common words.

TF-IDF is able to handle better variations in symptoms with many common words (persistent cough and chronic cough, for example), where the one-hot encoding would simply create two distinct columns. However, this transformation still doesn't enable to recognize similar words.

3) Word embeddings: Word embeddings are representation methods to sense the semantic proximity of words in a high dimension space. These methods are able to detect and remove stopwords, i.e., words that occur frequently but carry

minimal informational content.
In order to use these methods, we first need to transform our text data into tokens (small pieces of data: words, or more tuned tokens). We used the optimized tokenization of nltk.
Then we tried two different implementations of word embeddings : word2vec and doc2vec. In the word2vec approach, in order to have feature vectors of the same size, we computed the average of the vectors representing each word that describes a patient's symptoms. With doc2vec, we simply applied the model to the aggregated symptoms expressed in a single string. This is because doc2vec evaluates not just individual words, but also the context of entire documents or sentences.

4) BERT embeddings[2]: Transformers embeddings are currently the most accurate words representation. Unlike word2vec which generates context-independent embeddings for each word, Transformers provide context-sensitive representations, meaning the same word can have different embeddings based on its surrounding text; for example, the word "bank" would have different embeddings in "river bank" and "bank account". We chose to use BERT for our work. We first pre-processed our combined symptoms for each patient into tokens using the BERT tokenizer. Once tokenized, we applied the BERT model to the text, which outputs a vector representation for each token.

C. Data Split

In our data set, which includes three sets of symptoms for most diseases, we adopted a specific approach for splitting the data into training and test sets. This approach ensures that one-third of the data for each disease is reserved for testing, while two-thirds is used for training. A conventional split would have led to uneven representation of certain conditions in either the training or testing phases. The rationale behind this specific split is rooted in the nature of our task. It is crucial that the model is trained on a diverse range of examples for each disease to learn the underlying patterns effectively. However, it is equally important for the model to demonstrate its ability to generalize to new, unseen data. Moreover, it wouldn't make sense to evaluate the model on entirely unseen diseases, which would not reflect a real-world diagnosis scenario. In practice, a diagnosis model is expected to encounter variations of known diseases rather than completely new diseases it has never seen before. Therefore, our custom split is the most optimal strategy given our data.

D. Models

In our study, we initially employed a range of classification models to evaluate their performance for our comparative analysis. This included models like Random Forest, Linear Support Vector Classifier and Gaussian Naive Bayes. Each model was run on the dataset across different data representations, employing a three-fold cross-validation (3CV) strategy. This strategy was particularly chosen to match our dataset structure, where we had about three patients per disease, and to use all available data, reducing the risk of overfitting.
For initial model selection, we used the F1 weighted score as our primary metric, given its effectiveness in handling class imbalance and providing a balanced measure of the model's precision and recall. The top three models based on this metric were then selected for further tuning. The tuning process involved adjusting model parameters and fine-tuning them for each dataset type (using the right splitting described in section II-C)to achieve optimal performance. This step was crucial in understanding how each model responds to different data representations and preprocessing techniques.
After evaluating traditional models, our focus shifted to exploring the capabilities of more complex models like ANNs and CNNs. These models, known for their ability to capture deeper data patterns and relationships, were tested to see if they could outperform traditional machine learning models in our specific context. A simple ANN was first created, followed by a more intricate CNN, each fine-tuned and evaluated using the same performance metrics.

III. Results

To ensure the reproducibility of our results and maintain consistency across all experiments, we fixed a seed in our code for random number generation processes in all utilized libraries, including NumPy, TensorFlow, and pandas. This approach guarantees that the same initial conditions are used in each run, allowing others to replicate our findings accurately.
The figure 3 illustrates the performance of all tested models on the OHE dataset. This graph is a visual representation, highlighting each model's accuracy and f1 score both on train and test sets. The performance of models across other data types exhibited similar trends.
Our analysis identified three models with superior performance in terms of the F1 weighted score. These are:

1) Linear Support Vector Classifier (LSVC): Effective in high-dimensional spaces, excelling at finding optimal hyperplanes for classification.

2) Random Forest (RF): Demonstrates robustness using an ensemble of decision trees to handle complex datasets with numerous features.

3) Gaussian Naive Bayes (NB): Known for its simplicity and efficiency in probabilistic predictions.

We provide a table I detailing the performance and computation times of these top three models. This comparative overview allows us to assess the balance between performance and computational demands, highlighting the efficiency of each model. For each top model and for the two neural

Model	Fit Time	Test F1	Train F1
LinearSVC	1.190	0.191	0.980
RandomForestClassifier	5.495	0.168	0.981
GaussianNB	0.199	0.119	0.977

TABLE I: Model performance (F1 weighted score) and computation time comparison on OHE dataset with the 3CV and default parameters. The fit time corresponds to the time for fitting the estimator on the train set for each CV split.

networks, we generated plots that depict their performance across the five different data representations after tuning the hyperparameters. They present both training and test scores, which is crucial for identifying any potential overfitting issues. These plots 1 2 illustrate how each model adapts to different types of input data, providing insights into their versatility and robustness, and which data representations allow better classification results.

However, it is important to note that due to the computational intensity of the RF model, we were unable to generate complete results for all data representations. The extensive computation time, spanning several hours on our laptops, raises practical concerns about the feasibility of using the RF model in settings with limited computational resources. This consideration is particularly crucial in our target context of developing countries, where such resources may be scarce. In light of this, we have decided not to include incomplete results for the RF model in our detailed analysis. However, we do provide a summary table III, which includes the partial results we obtained for the RF model. Note that this model's performance seems worse than LSVC and GNB.

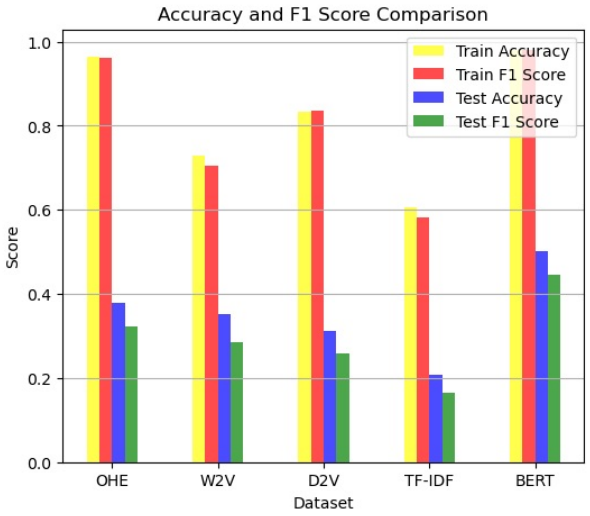

Fig. 1: Comparison of accuracy and F1 score for the LSVC model. Results are shown for both the train and test sets of the 5 different data representations.

IV. Discussion

A. Classification algorithms

In the evaluation of machine learning models for medical diagnosis, both the GNB and the Linear SVC exhibit strong signs of overfitting, as indicated by the disparity between training and test scores across all data representations. This common challenge underscores the complexity of medical data and the necessity for models to capture underlying patterns without adhering too closely to the particularities of the training set.

The LSVC, while not immune to overfitting, generally demonstrates a more robust performance compared to GNB, particularly in the W2V and BERT embeddings. This improved performance can be attributed to the known model's ability to establish good decision boundaries between classes, even in high-dimensional spaces. The strength of LSVC lies in its regularization framework, which effectively penalizes complexities in the model, thereby enhancing its generalization capabilities.

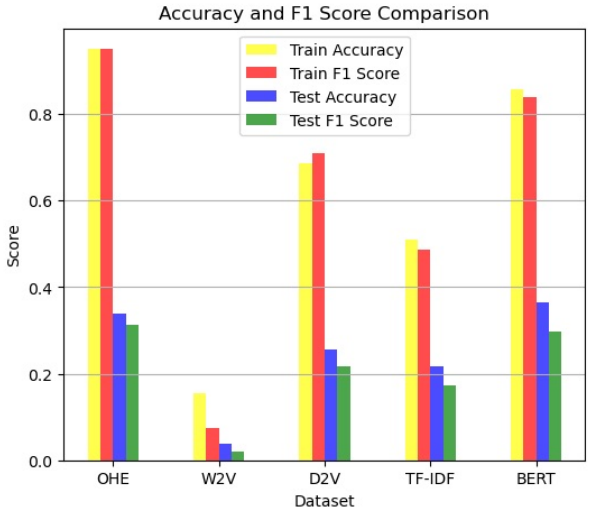

Fig. 2: Comparison of accuracy and F1 score for the GNB model. Results are shown for both the train and test sets of the 5 different data representations.

The GNB model, predicated on the assumption of feature independence, shows strong training results but falls short in testing scenarios. This suggests that the independence assumption does not hold well for the complex interdependencies present in medical diagnosis data.

In terms of their inherent advantages, the GNB model is typically lauded for its simplicity and speed, making it a viable option when computational resources are limited. Linear SVC, on the other hand, is more computationally intensive but is often preferred for its superior accuracy and robustness in diverse settings. Both models have their respective merits, and the choice between them may depend on the specific requirements of the application, such as the need for speed versus accuracy, the size and nature of the dataset, and the computational resources available. However, the overfitting observed and the overall modest scores achieved by both models point to the need for further refinement.

B. Neural networks

More sophisticated neural network architectures are theoretically capable of performing as well as or better than traditional machine learning models. This is due to their ability to learn complex, non-linear relationships in the data through multiple layers of abstraction. Neural networks excel in handling high-dimensional data, like the kind found in medical diagnosis tasks, where the relationships between inputs can be highly intricate and interdependent. Their deep learning

capabilities allow them to identify patterns that simpler models might miss. However, their superior performance is not guaranteed and depends on several factors, including: the quality and quantity of the training data, the network architecture and its suitability for the task, the tuning of hyperparameters and the prevention of overfitting through techniques like regularization and dropout.

The results depicted in the plots (see Appendix 4 and 5) illustrate the performance of two distinct neural network architectures. The ANN, utilizes a composition of Dense layers interspersed with Dropout layers to mitigate overfitting. The CNN incorporates also convolutional layers, batch normalization, and max pooling layers. The convolutional structure is particularly adept at recognizing spatial patterns within sentences describing symptoms, which could theoretically enhance the model's ability to understand and categorize medical conditions based on textual data.

However, the performance of both networks indicates that there is room for improvement. Notably, the networks underperform in comparison to traditional models, such as the LSVC, especially when evaluated on the W2V dataset. This suggests that despite the inherent advantages of neural networks and the efforts to refine the architectures and tune the hyperparameters, the current configurations do not fully harness these capabilities. This outcome suggests that there may be limitations to how much these particular neural network models can improve given the data and constraints we have.

C. Issues and improvements

Consequently, this opens the floor to LLMs like MEDITRON, which, with their advanced architectures and pretraining, hold significant promise for elevating performance and reducing overfitting. These models, designed to understand and generate human language, could offer considerable improvements in performance and robustness for medical diagnostic applications, surpassing the capabilities of both traditional algorithms and simpler neural network models.

A key limitation in our study is the limited dataset size, with only about three patients per disease class, which hindered the performance of all models. Both neural networks and traditional models struggled due to the need for large, diverse datasets to optimize and establish accurate patterns, respectively. This led to issues with generalization and a higher risk of overfitting. Attempts to use data augmentation were constrained by the complexities of medical terminology, making it challenging to generate contextually accurate synthetic data.

D. Breast Cancer Predictions

One possible explanation for the significant overfitting noted in our study was the limited amount of training data. To test this theory, we focused on the "Breast Cancer" condition, which had a higher representation in our dataset (24 patients, compared to only 3 for most other conditions). The predictions in the table below reveal not just an improved accuracy ($\approx 63\%$ for the best data transformations) for this particular condition, but also the effective performance of the word embeddings (Word2Vec, Doc2Vec, and BERT), underscoring their efficiency. Lastly, BERT predicted "Early-Stage Breast Cancer" three times, which also matched our true condition, demonstrating its context understanding. Those results were also observed with other over represented conditions.

OHE	**W2V**	**D2V**	**TFIDF**	**BERT**
Primary or secondary lung cancers	Metastatic Non-Castrate Prostate Cancer	Vitamin D deficiency, COVID-19	Potentially Curable Pancreatic Adenocarcinoma	Primary or secondary lung cancers
Breast Cyst	Breast Cancer	Breast Cancer	Urinary Tract Cancer	Breast Cancer
Breast Cancer	Breast Cancer	Breast Cancer	Ovarian cysts	Glaucoma
Opioid use disorder (OUD)	Anal Squamous Cell Cancers	Opioid use disorder (OUD)	Urinary Tract Cancer	Breast Cancer
Breast Cancer	Breast Cancer	Fibroadenor	Breast Cancer	Early-Stage Breast Cancer
Deep Venous Thrombosis...	Cellulitis	Varicose veins	Stage 4 prostate cancer	Breast Cancer
Breast Cancer	Breast Cancer	Breast Cancer	Cataract	Early-Stage Breast Cancer
Breast Cancer	Breast Cancer	Breast Cancer	active anky-losing spondylitis...	Early-Stage Breast Cancer

TABLE II: Predictions for Breast Cancer condition with Linear SVC and the various data transformations

V. Summary

This study conducted a comparative analysis of various machine learning models, including Linear SVC, Random Forest, Gaussian Naive Bayes, ANN and CNN, for medical diagnosis. We found that both traditional and neural network-based models tended to overfit, a challenge intensified by our dataset's limited size. Linear SVC was more effective, especially with W2V and BERT embeddings, compared to other models. The neural networks, despite their potential, did not perform optimally, indicating a need for further refinement. The study also underscores the promise of Large Language Models like MEDITRON for enhanced performance. Future efforts should focus on exploring diverse techniques, such as advanced embeddings, refined neural network architectures, and data augmentation, to improve the efficacy of machine learning in medical diagnostics.

VI. Ethics

Since we are working on an implementation for real world application, ethics are very crucial, especially for medical diagnosis application. Here we will focus on one particular question : Technological unemployment.

Technological unemployment is an important ethic issue for many digital applications. In medical context, it concerns populations in well developed nations, where a lot of people work in healthcare. Indeed, if one creates a tool with high and easy access for large scale, outperforming current accuracy, one might be highly concerned by this question. This scenario becomes more and more likely, as technology in AI is increasing at high rate, but we cannot control the severity of the consequences. The worst-case scenario would be that every people in healthcare would lose their job, but it more likely to reach a tradeoff between the initial situation and the worst-case scenario.

This risk is difficult to evaluate because it depends on so many factors. First the success of the final application, the appreciation of external users, and the society dynamics. Then we couldn't address any metrics for measurement. However, in comparison we can easily look at some graphs of AI popularity that are exploding today, and this might make us think about the likelihood of this risk in the future years. [3]

It was difficult for us to take this risk into account in our project since we work on the implementation and not on the deployment.

However, to illustrate the problematic and some ideas to take the risk into account let's put some context. The main goal of this project is to help people and save lives in underprivileged areas that considerably lack doctors to overcome the challenges of local medical services. In order to be ethically aligned here, further deployment could ensure that the tool is distributed to these specific areas by controlling its accessibility, but here we would face another ethic issue: accessibility.

From another point of view, areas potentially affected by technological unemployment are much more affluent and one could support that it is more benefits to save lives in such critical areas at the cost of losing jobs in affluent areas.

Overall, this is obviously a complex problematic for ethics, and the resolution is equally complex. The research work carries a part of responsibility towards its goals, but further deployment and applications should also proceed with caution and awareness of the ethical implications involved.

References

[1] "Meditron-70b: Scaling medical pretraining for large language models," 2023. [Online]. Available: https://arxiv.org/abs/2311.16079

[2] "Bert: Pre-training of deep bidirectional transformers for language understanding," 2019. [Online]. Available: https://arxiv.org/abs/1810.04805

[3] "Ai in 9 charts," 2022. [Online]. Available: https://hai.stanford.edu/news/state-ai-9-charts

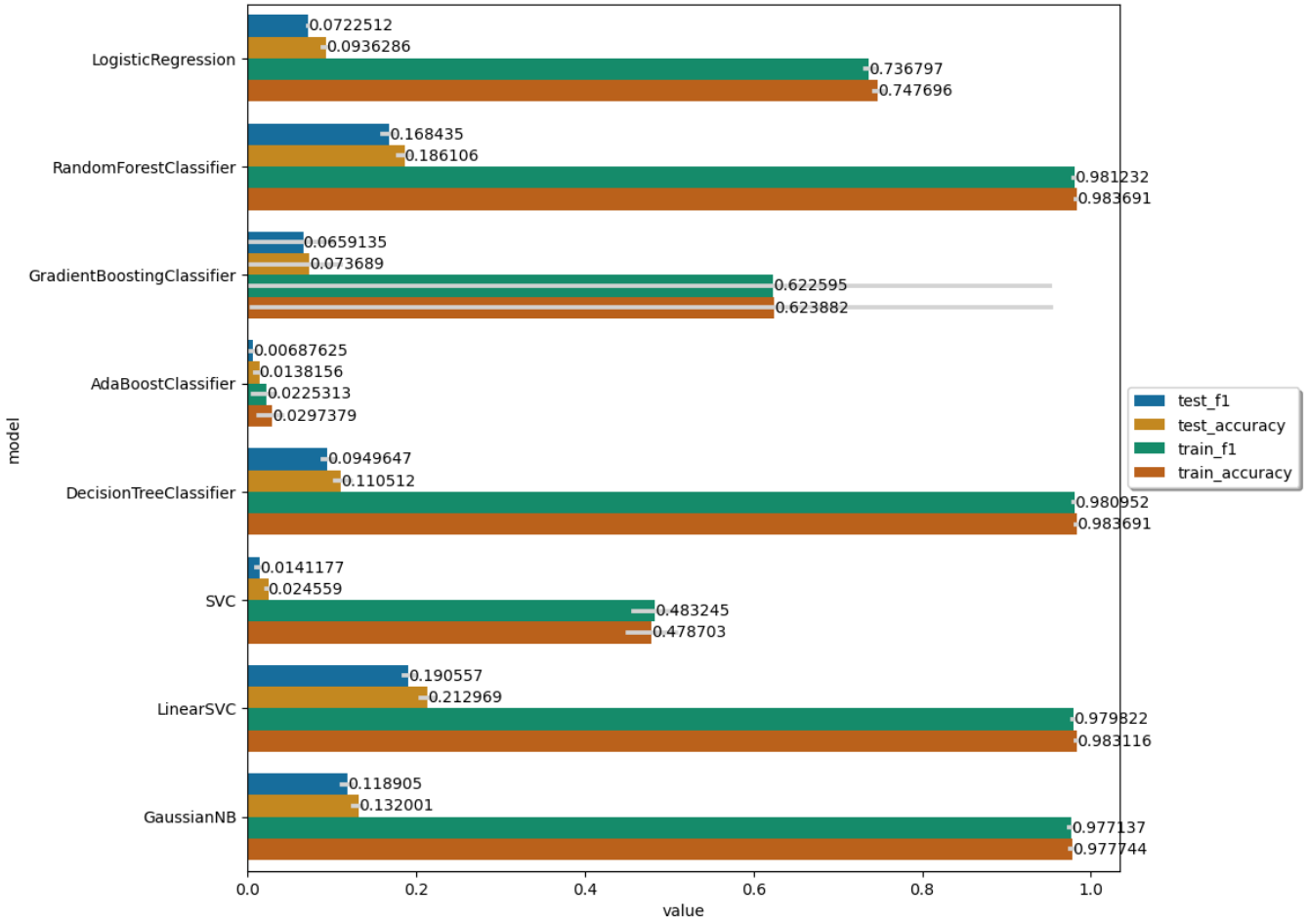

Fig. 3: Comparison of accuracy and F1 weighted score over a 3CV of the OHE for different traditional models with default parameters.

Data Representation	Test F1	Test Accuracy	Train F1	Train Accuracy
OHE	0.200	0.242	0.702	0.705
W2V	0.151	0.203	0.977	0.977
D2V	0.157	0.206	1.0	1.0
TF-IDF	0.131	0.178	0.636	0.642
BERT	N/A	N/A	N/A	N/A

TABLE III: Partial performance results of the Random Forest model across various data representations. Note that this model takes a long time to run and clearly overfit. With a higher number of estimators (here 300) and max depth (here 500) we could reach better performances but that would take even longer and overfit more.

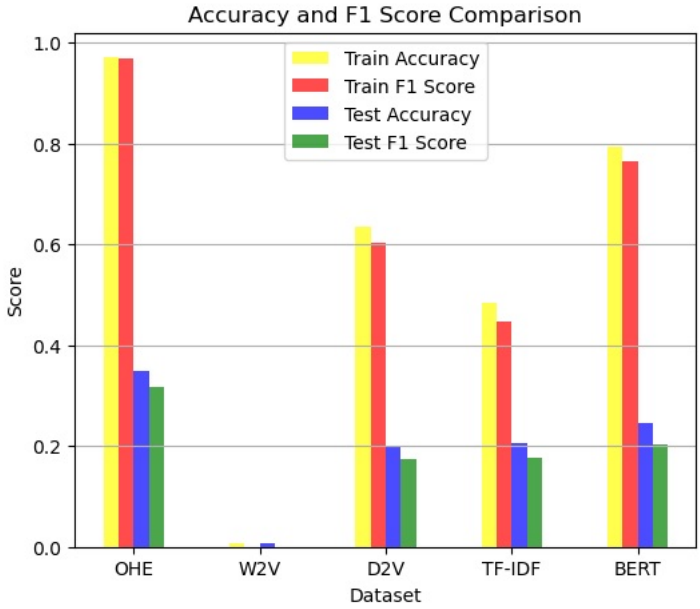

Fig. 4: Comparison of accuracy and F1 score for the ANN. Results are shown for both the train and test sets of the 5 different data representations.

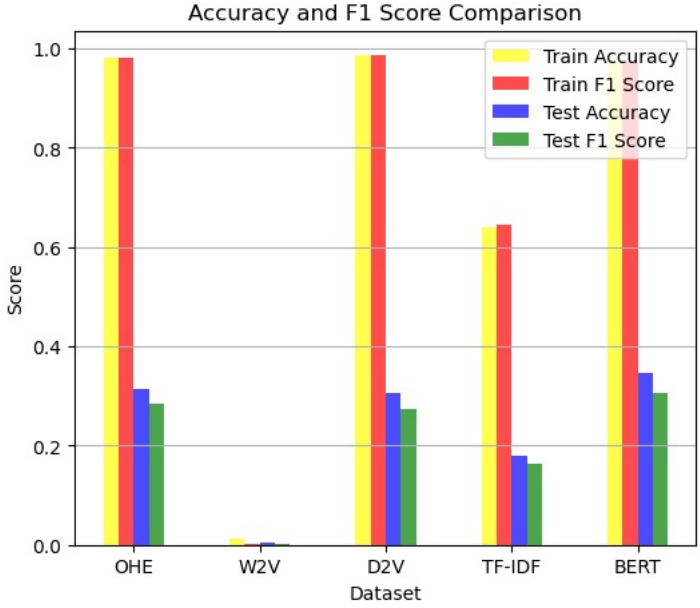

Fig. 5: Comparison of accuracy and F1 score for the CNN. Results are shown for both the train and test sets of the 5 different data representations.