BİLGİSAYAR BİLİMLERİ VE TEKNOLOJİLERİ DERGİSİ
JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGIES
http://dergipark.org.tr/tr/pub/bibted

Araştırma Makalesi

A Natural Language Processing-Based Turkish Diagnosis Recommendation System

Özlem Özcan Kılıçsaymaz*[1], Servet Badem[2]

Keywords:
AI-Based recommendation system, Turkish natural language processing, Long short-term memory, Recurrent neural network, Healthcare recommendation system

ABSTRACT

MD-Advisor is the abbreviation of "Medical Doctor–Advisor", a novel artificial intelligence-based (AI) recommendation system in healthcare. Moreover, the health-based recommender system is a decision-making tool that recommends appropriate healthcare information to patients and clinicians. It aims to minimize human error in the clinic and enhance patient safety by utilizing Natural Language Processing (NLP) methods and AI to diagnose certain cases that may otherwise be overlooked. In the fast-paced world of healthcare, it is essential for physicians to quickly and accurately diagnose patients to provide effective treatment. The MD-Advisor was developed to help healthcare professionals achieve this goal by speeding up the diagnostic process and presenting all possible conditions based on patient complaints. With this project, the methods of diagnosing the patient and then recommending the examination are completed quickly. Based on the data obtained from patient complaints that indicates the current health status of the patient; data preprocessing, labeling, and deep learning modeling techniques are used. The diagnostic codes used as labels for the diagnosis recommendation were obtained as output from the Recurrent Neural Networks (RNN) model. As a result of the study, the diagnosis proposal for the patient's complaints was successfully predicted with the applied RNN model approach.

Doğal Dil İşleme Tabanlı Türkçe Tanı Öneri Sistemi

Anahtar Kelimeler:
Yapay Zekâ Tabanlı Öneri Sistemi, Türkçe Doğal Dil İşleme, Uzun Kısa Süreli Bellek, Yinelemeli sinir ağı, Sağlık Öneri Sistemi

ÖZ

MD-Advisor, sağlık hizmetlerinde yapay zekâ tabanlı bir öneri sistemi olan "tıp doktoru – danışman" ifadesinin kısaltmasıdır. Ayrıca sağlık temelli öneri sistemi, hastalara ve klinisyenlere uygun sağlık hizmeti bilgileri için önerilerde bulunan bir karar alma aracıdır. MD-Advisor Projesi, doktorların hastalara teşhis koyarken izledikleri prosedürleri hızlandırmak ve olası tüm durumları kısa sürede doktora sunmak amacıyla geliştirilmiştir. Bu proje ile hastaya teşhis konulması ve sonrasında tetkik önerilmesi süreçleri çok hızlı bir şekilde tamamlanmaktadır. Böylece hasta doğrudan tedavi aşamasına geçmektedir. Hastanın mevcut sağlık durumunu gösteren hasta şikayetlerinden elde edilen verilere dayanarak; veri ön işleme, etiketleme ve derin öğrenme modelleme teknikleri kullanılmaktadır. Teşhis önerisi için etiket olarak kullanılan teşhis kodları, Tekrarlayan Sinir Ağları (TSA) modelinden çıktı olarak elde edildi. Çalışma sonucunda uygulanan TSA modeli yaklaşımı ile hastanın şikayetlerine yönelik tanı önerisi başarılı bir şekilde tahmin edilmiştir.

* **Responsible writer**

(ozlemozcank@gmail.com) ORCID ID 0000-0002-7282-512X
(servetbadem@gmail.com) ORCID ID 0000-0002-9883-3056

e-ISSN: 2717-8579

Arrival Date: 31/03/2022; Acceptance Date: 30/03/2023

1. INTRODUCTION

When a patient applied to a physician, approximately 100 thousand diseases are likely. The physician attempts to reduce these possibilities by taking patient's history, asking questions about their complaints, examining them, and ordering tests as required. After this process, clinicians make their decisions for diagnosis. AI-based recommendation systems, such as the MD-Advisor, aims to mimic this process by processing patient's complaints and reducing the number of potential diagnoses. These systems provide the physician with one or more of the most accurate diagnoses based on the patient's symptoms. The ultimate goal is to arrive at an accurate diagnosis in a timely and efficient manner.

These predictive technologies create an online model by integrating disease-related text and ontology features into intelligent algorithms. The use of recommendation systems has several advantages, such as reducing the workload for physicians and facilitating the inference of diagnoses made by other physicians based on patient complaints. The system trains itself with the past decisions of physicians through its learning process, offering the opportunity to compare its current recommendations with the physician's diagnosis. Furthermore, the system acts as an assistant to physicians, providing them with the control to make decisions independently.

The goal of the planned system is to minimize clinical decision error rates by presenting a diagnosis plan to the physician by integrating each phase of the program into the Electronic Healthcare Records (EHR) developed within the company. A clinical decision support system will be established that operates in harmony with the system and continuously trains itself based on physician feedback. This system will assist physicians in making critical decisions.

The data for the study was obtained from the Acıbadem Healthcare Group (AHG) hospitals. AHG is a healthcare institution offering services through various hospitals and medical centers in Turkey and all around the world.

It is widely recognized that oversights in the clinical field can result in serious problems that pose a significant risk to patient safety. Malpractices within the clinical process can lead to morbidity and even mortality. The MD-Advisor program aims to minimize human error rates in the clinic and enhance patient safety by utilizing Natural Language Processing (NLP) methods and AI to diagnose certain cases that may otherwise be overlooked.

To determine the data to be used in the project, data analysis was performed to determine the suitability of the data to be taken from the Hospital Information Management System (HIMS) screens for NLP. The EHR of the internal medicine department were used due to their comprehensive nature. The existing medical ontologies were evaluated, and the relevant classes, subclasses, and relations were identified for use in the ontology. Entities were detected with the help of Python. Many tools and libraries, such as NLTK, SpaCy, and Scikit Learn, which were created to apply NLP techniques and solve problems, were used in the text preprocessing stage.

The patient complaint data was preprocessed to make it usable for the project. During the preprocessing stage, the text was divided into tokens, and meaningless words and unnecessary characters were removed to obtain clean, understandable data. An RNN algorithm was created as a deep learning model for diagnosis prediction. As the output of this algorithm, a diagnosis recommendation is made based on the patient's situation. The study consists of three parts: introduction, methodology, and results. In the introduction, diagnostic methods and recommendation systems are described. In the methodology section, the project workflow is outlined, including data preprocessing, modeling, ontology, and Named Entity Recognition (NER) model. The results of the project were presented and analyzed in the last part, and the study is concluded.

2. BACKGROUND

Diagnosis in medicine refers to the process of determining the nature and circumstances of a medical condition through examinations and investigations. Furthermore, it refers to the decision reached from these processes as stated in the dictionary (“Definition of Diagnosis,” n.d.). In this context, diagnosis can be considered as a decision-making process under uncertainty, as stated in the article "How doctors diagnose diseases and prescribe treatments: an fMRI study of diagnostic salience" (Melo et al., 2017).

In the diagnostic process, physicians start by asking questions related to the patient's symptoms and complaints such as “What brings you here ?” and “What are your complaints ?”. Based on the patient's responses, the physician forms an initial impression of the potential disease. Subsequently, they start examination using medical tools according to the patient’s complaints and the area that causes the problem. If the problem is in the heart or lungs, they use a stethoscope; if the problem is in the ears of the patient, they use an otoscope, etc. Nevertheless, all the diagnostic tools cannot be mobile so they cannot be found in the doctor’s office. If the area causing the problem is not easily accessible, imaging tests such as Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) scans may be ordered. Despite advancements in investigation techniques, the process of gathering information through history taking still remains an important aspect of diagnosis. Nowadays, investigation techniques are improved, and doctors can detect diseases easier than ever. However, the part where doctors ask questions to their patients about the disease remains the same. Therefore, in this project, the focus is on utilizing

EHR to suggest possible diagnoses to physicians. EHRs capture the information recorded by physicians during patient interviews and are a valuable resource for diagnosis decision-making.

Recommender systems are designed to support medical care providers, particularly physicians, in the selection of appropriate diagnoses, treatments, medications, and other recommendations they may require. (Stark, Knahl, Aydin, & Elish, 2019) In recent decades, substantial amounts of data have been accumulated in clinical databases, including patients' health information, laboratory results, medical reports, treatment plans, and physicians' notes. As a result, the availability of digital information for patient-oriented decision-making applications has significantly increased. (Wiesner & Pfeifer, 2014). These systems are created to utilize patients' health information, such as EHR, to provide suggestions to healthcare providers, including doctors. It is important to note that these decision-making tools are not intended to replace healthcare providers but rather to enhance their diagnostic accuracy by automatically considering all possible diseases with the aid of machine learning and AI. Recommender systems can be beneficial in numerous areas of healthcare. For instance, the machine learning algorithm behind the system could be trained using images and patterns to make suggestions for imaging results such as MRI or CT scans. Additionally, the algorithm could be fed by physicians' written reports to incorporate verbal and written information from patients. Ultimately, recommender systems are tools to facilitate decision-making for healthcare providers, regardless of the type of data they are provided.

3. METHODS

Within the scope of the MD-Advisor Project, the workflow is divided into two primary categories: data preprocessing and modeling. The project also included ontology studies to implement the NER model for future use. Once the raw data was obtained, it was pre-suitable for the use and training of the model. To be used as input, two sections related to patients' complaints are taken. The system requires the disease data to be coded using the International Classification of Diseases (ICD-10) codes as the output for diagnosis estimation. The workflow of the project is illustrated in Figure 1.

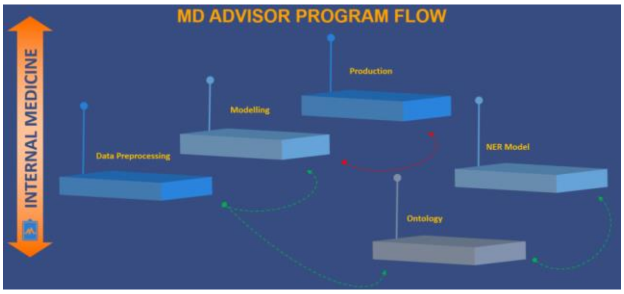


Figure 1. MD-Advisor Project Workflow

The model was selected as an RNN among deep learning models due to its recurrent structure. The deep learning model underwent training and was exported for clinical use after it successfully passed the testing and validation phases. A user interface was developed using the RNN model, based on the data from the hospital information system, to provide physicians with suggested diagnoses.

3.1. Data Source

The data set consists of three different columns. These three columns are respectively "COMPLAINTITLE", "COMPLAINTEXT" and "DIAGNOSIS". The first column usually contains a summary of the complaint in a single sentence. The second column is a paragraph containing details about the patient's complaint and history. The third column contains diagnoses in the form of ICD-10 codes.

The first column is filled when the doctors enter the patient's general complaints into the system while listening to the patient. More detailed information, such as the patient's past medical history, the medications they regularly use, his current illnesses, the details of their complaint, and family history, are included in the second column. The last column consists of diagnoses made by the physician considering the patient's complaints and history.

3.2. Data Preprocessing

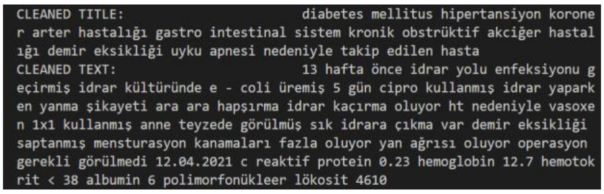
The raw data was obtained from the Relational Database Management System (RDBMS) of Acıbadem Hospitals.

Figure 2. MD-Advisor Dataset Characteristic

Therefore, the raw data was converted to JSON format for improved efficiency and ease of processing. Subsequently, the preprocessing of the data in the JSON format was initiated. Preprocessing was essential for the modeling and could impact the overall results of the project. This was due to the presence of various issues in the raw data, such as misspellings, empty rows, and special characters. The preprocessing steps involved the elimination of empty rows, conversion of all words to lowercase, tokenization of the dataset into individual words, and removal of special characters such as asterisks, symbols, and the 'less than' and 'greater than' symbols. Furthermore, all the abbreviations were detected and replaced with their meanings. Spell checking was also performed to correct misspelled, improperly joined, or separated words. Furthermore, the use of Regular Expressions (RegEx) was employed to locate units that had numbers before them, as these statements hold significance in the field of medicine. From the preprocessed data, specific columns that contain patients' complaints regarding their medical conditions and diseases were separated for further processing.

***"Sample Complaint Title** = dm htn cad gis copd patient followed for iron deficiency and sleep apnea."*

***"Sample Examination Notes =** uti 13 wk ago, had e.coli grown in urine culture. Have used cipro for five days, complaining of burning while urinating, frequent urination, occasional sneezing, urinary incontinence, used vasoxen 1*1 because of htn. seen in mother & aunt. heavy mens. bleeding with FE deficiency gis symptoms has flank pain, operation was not considered necessary. 12.04.2021 crp 0.23, hgb 12.7, hct <38, alb 6, pmnl 4610"*



```
CLEANED TITLE:                    diabetes mellitus hipertansiyon korone
r arter hastalığı gastro intestinal sistem kronik obstrüktif akciğer hastal
ığı demir eksikliği uyku apnesi nedeniyle takip edilen hasta
CLEANED TEXT:                     13 hafta önce idrar yolu enfeksiyonu g
eçirmiş idrar kültüründe e - coli üremiş 5 gün cipro kullanmış idrar yapark
en yanma şikayeti ara ara hapşırma idrar kaçırma oluyor ht nedeniyle vasoxe
n 1x1 kullanmış anne teyzede görülmüş sık idrara çıkma var demir eksikliği
saptanmış mensturasyon kanamaları fazla oluyor yan ağrısı oluyor operasyon
gerekli görülmedi 12.04.2021 c reaktif protein 0.23 hemoglobin 12.7 hemotok
rit < 38 albumin 6 polimorfonükleer lökosit 4610
```

Figure 3. Results of the Text After Being Preprocessed

"CLEANED TITLE: Patient followed up for diabetes mellitus, hypertension, coronary artery disease, gastrointestinal system disorders, chronic obstructive pulmonary disease, iron deficiency and obstructive sleep apnea."

"CLEANED TEXT: Patient had urinary tract infection 13 weeks ago, E.coli was grown in urine culture and used ciprofloxacin for five days. Complained of burning while urinating, frequent urination and occasional urinary incontinance when sneezing. Patient used to take vasoxen once a day for hypertension. Family history reveals that hypertension is also present in her mother and aunt. Patient has heavy menstrual bleeding and iron deficiency. Pain located in the flank. Operation was not considered necessary. Blood sample analysis in 12.04.2021 show; C-reactive protein: 0.23, hemoglobin: 12.7, hematocrit < 38, albumin: 6, polymorphonuclear leukocytes 4610"

3.3. Modeling

The MD-Advisor Program followed a structured approach in the modeling stage, comprising six steps. These steps were: tokenization, label encoding, the building of Multinomial Naïve Bayes Classifiers (MultinomialNB), the building of a Support Vector Classifier (SVC), the building of an RNN model, and evaluation of the model with different techniques through training, testing, and validation.

3.3.1. Tokenization

Tokenization is widely considered as the initial step in any NLP workflow and has a significant impact on the subsequent stages. A tokenizer divides unstructured data and text into discrete elements, referred to as tokens. Since machines cannot process words directly, tokenization converts words into numerical data structures, enabling the computer or machine learning pipeline to make complex decisions or perform actions (Menzli, 2022).

The MD-Advisor Program employs tokenization techniques for generating a large dataset for training the machine learning model. The tokenization process involves applying basic techniques to separate individual words, resulting in the creation of tokens. These tokens are then transformed into word vectors, allowing for effective training of the machine learning model, as machines are unable to comprehend words in their raw form.

3.3.2. Label encoding

Label Encoding is a method imported from the Scikit Learn library for normalizing labels in a dataset. This is a function that converts categorical data into numerical values and can be compared with each other. The label encoder technique assigns numerical values to categorical labels within an interval ranging from 0 to -1 and it assigns a unique value for each label type. In the process of label encoding, if a categorical label repeats, it will consistently be assigned the same numerical value as was previously assigned.

In the MD-Advisor Program, the target labels for the machine learning models were ICD-10 codes. To make analysis and comparison easier, the Label Encoder method was used to change these codes into numerical values. This step made it possible to use these numerical labels in the machine learning process and provide a standard way to analyze and model the data.

3.3.3. Multinomial Naïve Bayes Classifiers

Naïve Bayes is a method for classifying data based on probability. It is commonly used to categorize texts, based on the analysis of the words it contains. Unlike more complex AI-based approaches, Naïve Bayes offers a simpler solution for text classification.

The multinomial model is used for classifying non-numeric data. One of its benefits is that it is less complex than other models and can be trained with a smaller set of data, making it more efficient.

The goal of text classification is to sort the text into relevant categories. It evaluates the probability that a piece of text belongs to a particular group of similar texts. Each text is composed of multiple words that help to determine its meaning. A class is a label assigned to one or more texts that pertain to a similar subject.

The process of classifying a text involves analyzing the terms it contains and checking if they are found in other texts within the same class. This increases the likelihood that the text belongs to the same class as the previously classified texts.

The Naïve Bayes algorithm uses the Bayes theorem to categorize a text by assigning it a label. It calculates the probability of each label for a given text and assigns the class with the highest probability to that text.

The Naïve Bayes classifier is a group of algorithms that share a common assumption: that each feature being classified is independent of all other features. In other words, the presence or absence of a particular feature does not influence any other features.

Bayes theorem, named after Thomas Bayes, is a formula that calculates the probability of an event happening, taking into account prior knowledge of related conditions. It is calculated by the following formula:

where:

P(A): probability of case A
P(B): probability of case B
P(B|A): probability of A occurring when the estimator B is given

Default parameters are the preset variables of models. These variables are usually adjusted so that the models perform best. For this reason, these values have not been changed, and the context information of the MultinomialNB Classifiers model has been chosen as the default values. These values of the parameters are as follows:

Table 1. Parameters of MultinomialNB

Parameters	Default Values
alpha	1.0
force_alpha	False
fit_prior	True
class_prior	None

3.3.4. Support Vector Classifier

SVC is a machine learning method used for categorizing data into different classes. It's a supervised learning approach that transforms the input data into a higher dimensional space and then locates the best dividing line between the classes, called a hyperplane.

The hyperplane is picked in a way that maximizes the gap between the closest data points of each class, which are known as support vectors. This leads to a clear distinction between the classes and makes it possible to predict new data based on these classifications.

SVC is widely used in machine learning due to its capability to manage complex, non-linearly separable data and its reliability in providing accurate classifications even with limited training data. The Scikit Learn Library provides an implementation of SVC known as Sklearn SVC.

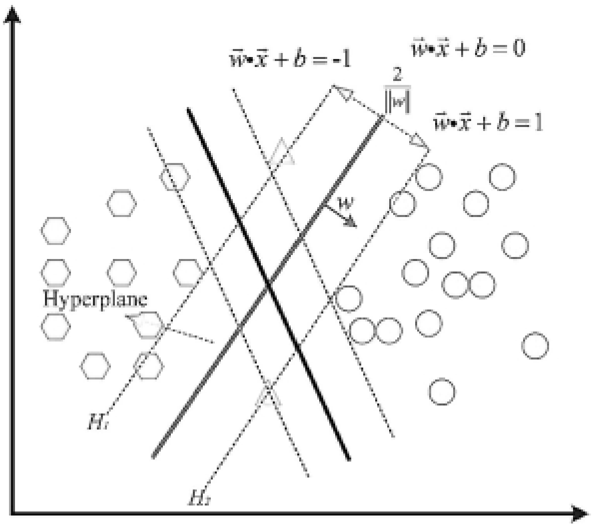


Figure 4. Illustration of the Support Vector Classifier

Since the default parameter values are determined as the best values for the performance of the model, they were not changed. Context information of the SVC model is selected as default values. These values of the parameters are as follows:

Table 2. Parameters of SVC

Parameters	Default Values
C	1.0
kernel	'rbf'
degree	3

gamma	’scale’
coef0	0.0
shrinking	True
probability	False
tol	1e-3
cache_size	200
class_weight	None
verbose	False
max_iter	-1
decision_function_shape	’ovr’
break_ties	False
random_state	None

3.3.5. Recurrent Neural Network

RNNs are a type of system that uses their previous output as an input in the next step, making them different from feedforward networks. This method allows them to have a “memory” and be useful when the input must be considered in a specific context to produce meaningful output. They are used to analyze and understand data in a specific order, such as texts, time-sensitive sensor data, and statistical data, which can not be effectively processed by feedforward networks.

In the MD-Advisor Program, the complaints indicating the patient's current state are the inputs for the model, and they are directly related to the diagnosis to be made for the patient. As a result, complaints are the most crucial factor in the diagnosis process. To accurately reflect this, the RNN model, which has a memory capability, was selected for use in the Project.

While creating the RNN model, embedding, which is a mapping method, was used. The use of embedding was chosen because it can reduce the complexity of categorical variables, allowing them to be effectively represented in the transformed space. The embedding size was chosen as 300. This helps to provide a meaningful representation of the categories.

In addition to embedding, the Long Short-Term Memory (LSTM) model was also used. LSTM is a type of RNN that has an extended memory capacity. It is commonly used for time series forecasting and serves as a building block for the layers of the RNN model. The LSTM assigns "weights" to data, allowing the RNN to effectively process new information, forget irrelevant information and give appropriate importance to data that affects the output.

The input length was chosen as 900, and the embeddings were set as trainable since there was no use of pre-trained embeddings.

In convolutional layers, Conv1D class is used to generate the output by creating a single spatial dimension from the inputs. The activation functions of these layers were selected as Rectified Linear Unit (ReLU).

A bidirectional class was used in the LSTM layer and the number of units was determined as 64.

Different length vectors temporally generated by LSTM cells were transformed into a single latent vector using the GlobalAveragePooling1D class. Then the output of the pooling layer is transmitted to the Dense layer.

The dense layer consists of 100 units and the number of units is chosen as 100.

By choosing the dropout value of 0.5, it was aimed to prevent the model from being overfit. With this method, some neurons are skipped and their weights are left unimportant.

Sparse Categorical Crossentropy is used to calculate the loss between labels and predictions. This loss function is chosen because the model will make multiclass classification and labels are encoded with Label Encoder.

Adam Optimizer, a stochastic gradient descent method, was chosen as the optimization method.

3.3.6. Model Evaluation

Model evaluation is an important process in which different statistical and mathematical methods are used to evaluate the performance and accuracy of the model. In this process, the model's predictions are compared with the actual results. The strengths and weaknesses of the classifications are determined. Based on these results, necessary changes and improvements are determined. Various criteria such as accuracy, precision, and recall are utilized to evaluate the model, depending on the particular problem and evaluation criteria.

3.3.6.1. Train/Test Split

The dataset used for the training and testing phases was obtained from AHG Hospitals. To evaluate the model correctly, 168184 rows of data were used for training. The validation set was determined as %25 of the training set. The test set was chosen as 55176 rows. Therefore, while 75.3% of the total data constituted the training and validation sets, 24.7% constituted the test set. It is important to provide the test set as well as the training data to determine whether the model is overfitting to the data set.

New patient entries are made every day in Acıbadem Hospitals and added to the database. Thus, new data is generated that the model has not been tested before. With this regular data flow, the model has the opportunity to go through new testing stages.

Furthermore, since the training/test split will be insufficient, the best parameters of the model can be found by using other model evaluation techniques.

3.3.6.2. Classification Metrics

There are four classification outcomes in which the model's predictions can be placed.

● True Positives: The examined sample belongs to the class in question and its estimation is made as "belongs".

● True Negatives: The examined sample does not belong to the class in question and is estimated as "does not belong".

● False Positives: The examined sample does not belong to the class in question and its estimation is made as "belongs".

● False Negatives: The examined instance belongs to the class in question and its estimated as "does not belong".

Table 3. Confusion Matrix

	Actual True	Actual False
Predicted True	True Positives	False Positives
Predicted False	False Negatives	True Negatives

The model was evaluated with four different proportional equations created using the specified outcomes.

● Accuracy: The ratio of correct classifications to all classifications. Accuracy is a measure of how well the model is doing in its predictions in general. A high accuracy value represents that the model has mostly correctly set up parameter selections and relationships.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{All\ Predictions}$$

● Precision: The ratio of true positive values to all predicted positive values. Precision can be considered as a quality determining factor. Examines how much of the predictions are relevant. When higher precision is achieved, instances are predicted in a more relevant manner.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

● Recall: The ratio of true positive values to all actual positive values. Recall can be considered as a quantitative determinant. It examines how much of the instances return as relevant in reality. Obtaining a high recall value indicates that the most relevant instance values have been estimated.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

● The F1 score: It is a tool used to assess the accuracy of a classification model on a specific dataset. It combines the precision and recall metrics into one single measure and is particularly useful for evaluating performance on imbalanced datasets. The F1 score is calculated as the harmonic mean of precision and recall, which involves taking the average of these two metrics.

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

The F1 score is high when both of these metrics are high and the score is low when the metrics are low. When a metric is high and the other is low, the F1 score is medium.

3.4. Ontology

An ontology is a model that defines concepts and relationships between them in a specific field. It provides a structured way to access and understand information and ensures accuracy in retrieving meaning from that information. The underlying logic behind ontologies is that in any field, the terms used are interconnected. In the medical field, for example, an ontology would connect diseases, medications, and treatment procedures. (Adelkhah, Shamsfard, & Naderian, 2019).

In the context of the MD-Advisor Program, an ontology is used to develop an NER model to assist doctors in finding relevant information about medical terms and their relationships. Having such a huge ontology in the medical field helps healthcare providers to approach diseases, medications, treatments, etc. professionally.

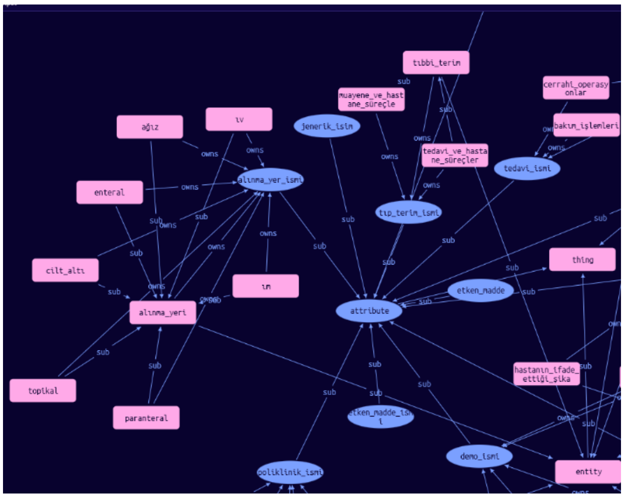


Figure 5. Ontology

3.4.1 Named Entity Recognition

Furthermore, NER is widely used in NLP to categorize words into predefined categories. (Li, 2018).

In the MD-Advisor Program, NER is used to identify the medical entities in patient notes when they are being recorded in the system. When the NER model is activated, it highlights any detected medical entities and identifies their predefined category, such as diagnosis, medication, treatment method, time expression, etc.

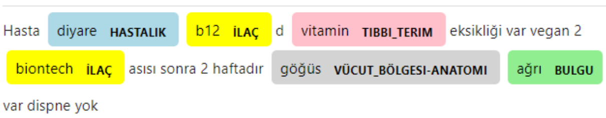


Figure 6. Named Entity Recognition Original Turkish Text

" The patient has diarrhea DIAGNOSIS b12 DRUG vitamin MEDICINE_TERM d deficiency vegan 2 weeks after biontech DRUG vaccine has chest BODY_REGION_ANATOMY pain FINDINGS no dyspnea"

4. RESULTS

For the MD-Advisor Program, 168184 lines of internal medicine data were used as the training set for the RNN model with LSTM cells. This set contained the examination data of 167916 patients.

25% of the validation data was separated from the training set. The model was trained with 150 epochs and patient complaints were used as input. First of all, the results obtained here were examined. Accuracy was achieved as 0.8020 and the loss of accuracy after testing was 0.7500.

Owing to the regular data flow from hospitals, a test set was created with new data that the model never encountered. The test set included internal medicine data from 01.11.2022 to 01.01.2023. This set corresponded to 32.8% of the training set and 24.7% of the total dataset.

Then, this newly created test set of 55176 lines was given to the RNN model and the outputs were examined. Accuracy 0.6145, precision 0.3163, recall 0.4460 and F1 score 0.3487 were obtained.

To compare the performance of the model and make a selection, two more models were trained and their outputs were examined, apart from the RNN model.

According to the results obtained from MultinomialNB Classifiers, accuracy was 0.2894, precision 0.0812, recall 0.1353, and F1 score 0.0916.

After the SVC model was tested with the specified test data, its accuracy value was recorded as 0.2547, precision 0.1487, recall 0.2401, and F1 Score 0.1651.

Table 4. Comparison of the Results of the Models

	RNN	Naïve Bayes	SVC
Accuracy	0.6145	0.2894	0.2547
Precision	0.3163	0.0812	0.1487
Recall	0.4460	0.1353	0.2401
F1 Score	0.3487	0.0916	0.1651

5. DISCUSSION

As in every field, applications of machine learning and deep learning models are increasing in order to create clinical decision support and prediction systems in the field of healthcare. Based on patient complaints, MD-Advisor makes diagnostic recommendations coded with ICD-10.

Diagnostic recommendations provide full code estimates. The full code estimate gives an output of five characters. The first three characters denote the category of the disease, and the last two characters give more specific details about the disease. In such a clinical classification problem, a person may have one of several disease classes or may have several diseases at the same time. For this reason, the categorization made is both multiclass and multilabel classification.

Various models and methods have been used in the MD-Advisor Program, from machine learning and deep learning algorithms to evaluation metrics.

Two machine learning and one deep learning model were built to select the best-performing model that MD-Advisor will use.

5.1. Machine Learning

Among the machine learning algorithms, MultinomialNB Classifiers and SVC models were developed.

According to the results obtained, it was observed that the performance of the SVC model was better than the MultinomialNB, except for the Accuracy value. The accuracy of the SVC model was 0.2547, lower than the MultinomialNB with 0.2894 accuracies.

The difference between precision, recall, and F1 score values of SVC and MultinomialNB models is greater than the difference between accuracy values. Therefore, the low accuracy of SVC can be ignored and it can be said that it performs better than MultinomialNB.

In prediction systems in the field of health, it is important to what extent the diseases of individuals can be predicted correctly. The Recall value gives the rate of how many of the patients with the disease are diagnosed with the specified disease. For this reason, the Recall value is among the important metrics to be considered. The fact that the Recall value of the SVC model is about two times the value of the MultinomialNB shows that the performance of the SVC is better.

5.2. Deep Learning

RNN, which contain LSTM layers, have been chosen as the deep learning algorithm. This model with memory makes it possible to perceive and process the concept of the complaints entered.

The accuracy, precision, recall, and F1 score values of the RNN model were obtained as 0.6145, 0.3163, 0.4460, and 0.3487, respectively. It has been observed that these values obtained from the RNN model are better than the values of SVC and MultinomialNB machine learning models.

The accuracy of the RNN model is about two times the accuracy of MultinomialNB and 2.5 times the accuracy of the SVC model. In clinical systems, the recall value, which shows the rate of a correct positive diagnosis, is also crucial. This value is approximately three times the recall of MultinomialNB and two times the recall of the SVC model. These results led to the selection of the RNN model for the MD-Advisor Program.

As the MD-Advisor RNN model is used by physicians, it will continue to train itself and provide better performance. These initial results from evaluation metrics will increase as usage increases and continues.

Some diagnoses entered by physicians may differ from the actual diagnosis of the patient, or the patient's complaints may be very brief. For some reports, such as military service reports, a random diagnosis can be entered. Since there is no routine control head for assesments such as check-ups, upper respiratory tract infection (J06.9, ICD-10) is usually entered. All these factors negatively affect the performance of the model. If the stated conditions are improved, an increase in performance is also expected.

6. CONCLUSION

The MD-Advisor Program aims to enhance the accuracy and quality of diagnosis and treatment within the AHG. MD-Advisor, which is supported by AI, will assist physicians in making accurate diagnoses by addressing their needs and providing diagnostic support.

The higher precision in predictions made by the AI system among the pool of diagnoses will result in more accurate patient treatments by providing more accurate diagnoses.

MD-Advisor, which currently offers 80% accuracy on diagnosis recommendations, will provide physicians with more accurate results over time thanks to the continuous learning feature of its AI-supported systems. It is expected to effectively recognize difficult-to-detect cases, particularly in rare diseases, resulting in improved accuracy of diagnosis selection. Not only the practices of a single physician but the treatment protocols of all Acıbadem physicians will be examined and presented to physicians in the form of appropriate treatment protocols. In this way, a good practice among Acıbadem physicians will be developed and disseminated through MD-Advisor supported by AI.

MD-Advisor, an AI-powered clinical decision support system, has been acquired by AHG to meet its specific needs and demands, reflecting the global acceptance and utilization of such systems in healthcare. With the implementation of the MD-Advisor, there will be a dynamic process for the system to develop and become widespread, along with more demands and needs that will arise as a result of the use of AI-supported systems.

The successful use of MD-Advisor, developed by AHG's in-house resources, will be the first step for the organization to play a pioneering role in healthcare technologies and AI.

REFERENCES

Adelkhah, R., Shamsfard, M., & Naderian, N. (2019). The ontology of natural language processing. *5th International Conference on Web Research (ICWR)*, 128-133.

Kaur, R., Ginige, J.A., & Obst, O. (2021). A systematic literature review of automated ICD coding and classification systems using discharge summaries. *ArXiv, 2107. 10652.*

Ma, F., Chitta, R., Zhou, J., You, Q., Sun, T., & Gao, J. (2017). Dipole: diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. *Proceedings of the*

23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Melo, M., Gusso, G.D.F., Levites, M., Massad, E., Lotufo, P.A., Zeidman, P., . . . Price, C.J. (2017). How doctors diagnose diseases and prescribe treatments: an fMRI study of diagnostic salience. *Scientific Reports*, *7*(1), 1304.

Plisson, J., Lavrač, N., & Mladenic, D. (2004). A rule based approach to word lemmatization.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *ArXiv, 1609.04747*.

Stark, B., Knahl, C., Aydin, M., & Elish, K. (2019). A Literature Review on Medicine Recommender Systems. International Journal of Advanced Computer Science and Applications, 10(8).

Wiesner, M., & Pfeifer, D. (2014). Health Recommender Systems: Concepts, Requirements, Technical Basics, and Challenges. International Journal of Environmental Research and Public Health, 11(3), 2580–2607.

Brownlee, J. (2022). Your first deep learning project in python with keras step-by-step. *Machine Learning Mastery*. Retrieved from https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/.

Brownlee, J. (2021). How to choose an activation function for deep learning. *Machine Learning Mastery.* Retrieved from https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/.

Brownlee, J. (2019). A gentle introduction to cross-entropy for machine learning. *Machine Learning Mastery.* Retrieved from https://machinelearningmastery.com/cross-entropy-for-machine-learning/.

Brownlee, J. (2017). How to visualize a deep learning neural network model in keras. *Machine Learning Mastery.* Retrieved from https://machinelearningmastery.com/visualize-deep-learning-neural-network-model-keras/.

Brownlee, J. (2017). Gentle introduction to the adam optimization algorithm for deep learning. *Machine Learning Mastery.* Retrieved from https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/.

Brownlee, J. (2016). 5 step life-cycle for neural network models in keras. *Machine Learning Mastery.* Retrieved from https://machinelearningmastery.com/5-step-life-cycle-neural-network-models-keras/.

Brownlee, J. (2016). Multi-class classification tutorial with the keras deep learning library. *Machine Learning Mastery.* Retrieved from https://machinelearningmastery.com/multi-class-classification-tutorial-keras-deep-learning-library/.

Jain, V. (2019). Everything you need to know about “activation functions” in deep learning models. *Towards Data Science.* Retrieved from https://towardsdatascience.com/everything-you-need-to-know-about-activation-functions-in-deep-learning-models-84ba9f82c253.

URL-1: https://www.dictionary.com/browse/diagnosis [Access Date: January 2023]

URL-2:https://www.healthit.gov/faq/what-electronic-health-record-ehr [Access Date: January 2023]

URL-3: Anonymous, (2020). How Does the Gradient Descent Algorithm Work in Machine Learning? https://github.com/visionatseecs/keras-starter/blob/main/keras_intro_mlp.ipynb [Access Date: January 2023]

URL-4: https://www.analyticsvidhya.com/blog/2020/10/how-does-the-gradient-descent-algorithm-work-in-machine-learning/ [Access Date: January 2023]

URL-5: https://tutorialspoint.com/deep_learning_with_keras/deep_learning_with_keras_tutorial.pdf [Access Date: January 2023]

URL-6:https://deepai.org/machine-learning-glossary-and-terms/softmax-layer [Access Date: January 2023]

URL-7: https://deepnotes.io/softmax-crossentropy [Access Date: January 2023]

URL-8: Li, S. (2018). Named Entity Recognition with NLTK and SpaCy. https://towardsdatascience.com/named-entity-recognition-with-nltk-and-spacy-8c4a7d88e7da [Access Date: January 2023]

URL-9: Menzli, A. (2022). Tokenization in NLP: Types, Challenges, Examples, Tools. https://neptune.ai/blog/tokenization-in-nlp [Access Date: January 2023]

URL-10: Arnx, A., (2019, Jan 13). First neural network for beginners explained (with code).
https://towardsdatascience.com/first-neural-network-for-beginners-explained-with-code-4cfd37e06eaf
[Access Date: January 2023]

URL-11: Nielsen, M., (2019). Neural Networks and Deep Learning. Neural Networks and Deep Learning,
http://neuralnetworksanddeeplearning.com
[Access Date: January 2023]

URL-12: Trehan, D. (2022). Gradient Descent Explained.
https://towardsdatascience.com/gradient-descent-explained-9b953fc0d2c
[Access Date: January 2023]

URL-13: Srivastava, K., (2021, Jan 21). Classification – Let's understand the basics.
https://towardsdatascience.com/classification-lets-understand-the-basics-78baa6fbff48
[Access Date: January 2023]

URL-14: Roman, V., (2019). Supervised Learning: Basics of Classification and Main Algorithms. Towards Data Science.
https://towardsdatascience.com/supervised-learning-basics-of-classification-and-main-algorithms-c16b06806cd3
[Access Date: January 2023]

URL-15: Saxena, S. (2021). Binary Cross-Entropy/Log Loss for Binary Classification.
https://www.analyticsvidhya.com/blog/2021/03/binary-cross-entropy-log-loss-for-binary-classification/
[Access Date: January 2023]

URL-16: Godoy, D. (2018). Understanding Binary Cross-Entropy / Log Loss: A Visual Explanation.
https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a
[Access Date: January 2023]

URL-17: Sharma, A., (2017). Understanding Activation Functions in Neural Networks. The Theory of Everything.
https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0
[Access Date: January 2023]

URL-18: Sharma, P. (2020). Keras Optimizers Explained with Examples for Beginners.
https://machinelearningknowledge.ai/keras-optimizers-explained-with-examples-for-beginners/
[Access Date: January 2023]

URL-19: Gomez, R.,(2018). Understanding Categorical Cross-Entropy Loss, Binary Cross-Entropy Loss, Softmax Loss, Logistic Loss, Focal Loss, and all those confusing names.
https://gombru.github.io/2018/05/23/cross_entropy_loss/
[Access Date: January 2023]

URL-20: Wambui, R. (2022). Cross-Entropy Loss and Its Applications in Deep Learning.
https://neptune.ai/blog/cross-entropy-loss-and-its-applications-in-deep-learning
[Access Date: January 2023]