Interdisciplinary Journal of Information, Knowledge, and Management

An Official Publication of the Informing Science Institute
InformingScience.org

IJIKM.org

Volume 19, 2024

A Learn-to-Rank Approach to Medicine Selection for Patient Treatments

Maher Farouqa — Princess Sumaya University for Technology, Amman, Jordan — mah20198075@std.psut.edu.jo

Mohammad Azzeh* — Princess Sumaya University for Technology, Amman, Jordan — m.azzeh@psut.edu.jo

* Corresponding author

Abstract

Aim/Purpose This research utilized a learn-to-rank algorithm to provide medical recommendations to prescribers. The algorithm has been utilized in other domains, such as information retrieval and recommender systems.

Background Ranking the possible medical treatments according to diagnoses of the medical cases is very beneficial for doctors, especially during the coding process.

Methodology We developed two deep learning pointwise learn-to-rank models within one prediction pipeline: one for predicting the top possible active ingredients from disease features, the other for ranking actual medicines codes from diseases and the ingredients features.

Contribution A new learn-to-rank deep learning model has been developed to rank medical procedures based on datasets collected from insurance companies.

Findings We ran 18 cross-validation trials on a confidential dataset from an insurance company. We obtained an average normalized discounted cumulative gain (NDCG@8) of 74% with a 5% standard deviation as a result of all 18 experiments. Our approach outperformed a known approach used in the information retrieval domain in which data is represented in LibSVM format. Then, we ran the same trials using three learn-to-rank models – pointwise, pairwise, and listwise – which yielded average NDCG@8 of 71%, 72%, and 72%, respectively.

Recommendations for Practitioners The proposed model provides an insightful approach to helping to manage the patient's treatment process.

Accepting Editor Geoffrey Z. Liu │Received: March 12, 2024│ Revised: July 22, August 8, August 27, August 30, 2024 │ Accepted: August 31, 2024.

Cite as: Farouqa, M., & Azzeh, M. (2024). A learn-to-rank approach to medicine selection for patient treatments. *Interdisciplinary Journal of Information, Knowledge, and Management, 19,* Article 30. https://doi.org/10.28945/5371

Recommendations for Researchers	This research lays the groundwork for exploring various applications of data science techniques and machine learning algorithms in the medical field. Future studies should focus on the significant potential of learn-to-rank algorithms across different medical domains, including their use in cost-effectiveness models. Emphasizing these algorithms could enhance decision-making processes and optimize resource allocation in healthcare settings.

Impact on Society	This will help insurance companies and end users reduce the cost associated with patient treatment. It also helps doctors to choose the best procedure and medicines for their patients.

Future Research	Future research is required to investigate the impact of medicine data at a granular level.

Keywords	learn to rank, medicine ranking, decision making

INTRODUCTION

The medical field has long been a cornerstone of research due to its vital role in promoting human health and well-being (Subrahmanya et al., 2022). Medical professionals and insurance claim coders are responsible for inputting detailed medical claims into healthcare management systems (Y. Yang et al., 2022). This task requires precise coding of diagnoses for each patient case to ensure that all stakeholders involved can accurately interpret and utilize the information. Additionally, it is crucial to document subsequent treatments, including medications, procedures, radiological tests, and physiotherapy sessions. A major challenge within the healthcare sector is the potential for physicians to select inappropriate or suboptimal treatments, which can damage their professional reputation, jeopardize patient health, and negatively impact insurance companies' financial stability. Enhancing the accuracy of physicians' treatment decisions by narrowing down and prioritizing the most relevant treatment options is essential to mitigate this issue. This need has driven the development of decision-support mechanisms. It is important to note that contracts do not influence the actual treatment decisions, which the physician ultimately determines. However, contractual agreements may sway a physician's choice of medication brand, especially when alternative options have similar therapeutic properties. Factors such as medication cost and quantity can significantly influence these choices, aiding physicians in quickly selecting the necessary medications. Many patients visiting doctors have private insurance, and physicians often have contracts with insurance companies that impose certain restrictions on their choice of medication brands. While not affecting the treatment itself, these contracts may influence the selection of the most suitable medication brand that meets the patient's medical needs and the contractual terms with the insurance company. Therefore, it is imperative to tailor treatment options to each patient's case, as multiple treatments may be appropriate for the same condition, depending on the physician's clinical judgment (Reig et al., 2022).

Data science approaches are rapidly advancing and demonstrating significant impact across various domains, including healthcare. We believe these methods hold great potential and promise within the healthcare sector. This paper focuses on leveraging a specialized branch of data science, specifically learning to rank (LTR), to prioritize the most effective medications for particular diseases. Numerous studies have explored the application of machine learning and deep learning techniques in the medical field, addressing diverse areas. One notable area is international classification of diseases (ICD) coding, where clinical texts are classified using higher-level coded medical concepts (Hosseini et al., 2018). Additionally, future diagnosis prediction has been a key focus, where temporal historical visit data is used to predict potential future diseases for patients, as demonstrated by Meng et al. (2020) and Ma et al. (2017). Another critical application of data science in healthcare is predicting and ranking treatments based on their relevance, effectiveness, and potential side effects (Gerdes et al., 2021).

In this context, researchers have applied various machine learning and deep learning algorithms to classification, regression, and LTR problems, aiming to identify the most appropriate treatments – including medical procedures and medications – for diagnosed cases. These decisions are informed by numerous relevant factors, including the disease characteristics, symptoms, patient profile, and other crucial determinants (Levy et al., 2022).

The vast number of diseases and treatments, including laboratory tests, radiology procedures, and medications, present significant challenges (Rehman et al., 2022). In this paper, we focus specifically on medications due to their critical role in disease treatment and their substantial share of the overall treatment landscape, both in terms of frequency and cost. Real-world datasets often suffer from data quality issues, which can hinder the effectiveness of predictive models used for treatment ranking (Norori et al., 2021). These issues may include the absence of critical features, such as accurate symptom information, key aspects of patient profiles, and individual doctors' preferences. However, these datasets also offer a rich source of historical data that can be leveraged to extract valuable insights and enhance predictive model performance. Our study utilizes an 18-month dataset of historical doctor visits, including information on prescribed medications. By analyzing this historical data, we can calculate the frequency of medication usage for specific diseases. This frequency data can then be used as labels in LTR approaches, thereby improving the ranking of medications based on their prescription frequencies (Electronic Health Solutions, 2019).

These needs serve as the primary motivations for this study. LTR approaches could be pivotal in aiding doctors in rapidly and accurately selecting the most appropriate medications that meet patient-specific requirements and insurance company standards. This paper aims to experiment with and assess the feasibility of LTR methods for ranking the top medical treatments per disease. Such an approach is expected to provide substantial benefits to doctors, patients, medical coders, and insurance companies by enabling a faster and more precise selection of medications that satisfy all stakeholders involved. We believe this paper makes several contributions that could positively impact the quality of medical services and reduce the time required for data collection. By ranking the most applicable medications, the accuracy of treatment coding could be enhanced, and the process of assigning treatments by system users, including doctors and coders, could be significantly expedited.

In summary, we summarize the main contributions of this paper into the following points:

- Using data science approaches, such as LTR, and examining their effectiveness on real-world datasets.
- Proposing a new approach that is based on LTR pointwise for ranking medicines.
- Comparing our approach with a known benchmark approach in which three different LTR models – pointwise, pairwise, and listwise – have been tested on the same dataset.

This paper is structured into seven sections. The Introduction section introduces the problem and outlines the objectives, providing an overview of the research problem, goals, and the overall structure of the paper. The Background section delves into the background of the issue, including an overview of standardized medical diseases and treatment codes, as well as a comprehensive description of the claims handling process within the insurance sector, highlighting the challenges related to data quality and flow. The Literature review section reviews related work in the field. In the Methodology section, we discuss the characteristics of the datasets, the proposed approach, data representation and preprocessing methods, LTR techniques, and relevant evaluation metrics. The Implementation section presents the implementation details, results, and a discussion of the findings. The Threats to validity section addresses potential threats to the validity of our study. The conclusion section concludes the paper, summarizing the key insights and contributions.

BACKGROUND

To gain a clearer understanding of the problem, it is essential to provide some contextual details. This paper focuses on applying LTR techniques to medical data, which necessitates a thorough explanation of key features and labels, including the International Classification of Diseases (ICD) and Current Procedural Terminology (CPT) code lists. Additionally, the process of managing medical insurance claims is discussed, along with the relevant medical data formats and workflows associated with these claims.

STANDARD MEDICAL CODED LISTS (ICD AND CPT)

For a better understanding of the work presented in this paper, it is crucial to elaborate on the standardized medical coding systems utilized in the dataset. These codes play a significant role in structuring the data and ensuring consistency. The standardization of disease classification began in the 1850s (Scichilone & Giannangelo, 2013). The ICD-9, the ninth revision of the ICD, was widely adopted globally until 2015. Subsequently, the World Health Organization (WHO) introduced ICD-10, which was endorsed by the World Health Assembly in 1990 and released for international use in 1994. The United States implemented ICD-10 in October 2016, albeit with delays due to the complexity, cost, and need for extensive training and system updates. The transition aimed to register new diseases and injuries while enhancing the standardization of coding by unifying its structure and allowing for more detailed descriptions when necessary. ICD-10 includes 69,823 disease diagnosis codes and 71,924 procedure codes, with lengths varying from 3 to 7 digits depending on the level of detail.

The structure of an ICD-10 code is as follows: the first digit is a non-case-sensitive letter (A-Z, excluding "U") representing the disease chapter, the second digit is a number, and digits three through seven are alphanumeric, providing further specificity. For example, "H" represents diseases related to the eye and ear. The remaining digits provide subcategories, and in some cases, an extension code is used, particularly for obstetrics, injuries, and external causes of injuries (Subotin & Davis, 2014). For instance, the code "S06.0x1A" refers to an injury classified as "Concussion with loss of consciousness of 30 minutes or less, initial encounter" within the Injuries chapter, identified by the letter "S" and further categorized under "Intracranial injury" with the diagnostic code "S06." It's important to note that the detailed nature of ICD-10 codes allows for distinctions such as laterality (the specific side of the body) and the differentiation between a new occurrence of a disease and its recurrence. For example, different codes are assigned to the same eye disease depending on whether it affects the right or left eye. This granularity can impact prediction models, as the same disease might be represented by different features due to varying codes. However, the influence on prediction accuracy depends on the specific use case and whether the objective is to differentiate between new cases and recurrences.

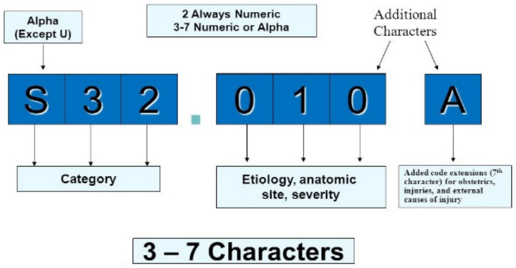

Alpha (Except U)
2 Always Numeric
3-7 Numeric or Alpha
Additional Characters
S
3
2
.
0
1
0
A
Category
Etiology, anatomic site, severity
Added code extensions (7th character) for obstetrics, injuries, and external causes of injury
3 – 7 Characters

Figure 1. ICD-10 code structure

Current procedural terminology (CPT) is a standardized coding system created by the American Medical Association (AMA) to classify outpatient medical treatments performed by physicians. It uses a 5-digit format, which can be numeric or alphanumeric depending on the CPT category, to ensure that medical and surgical services are uniformly coded across various electronic health record (EHR) systems, reimbursement systems, and other platforms. This standardization allows for seamless integration of procedural data between systems, reducing the need for complex, time-consuming, and error-prone mapping activities. The use of CPT codes is crucial because medical reimbursement entities, such as insurance companies and national healthcare programs, strictly enforce rules that reject improperly coded procedures, incorrect treatments, or inconsistencies that could negatively impact the reimbursement process. Additionally, accurate coding is essential for reliable reporting and analysis, which directly influences effective decision-making in healthcare.

The CPT list is updated annually to reflect new medical and technological advancements and emerging needs. CPT is organized into three categories: Category I, Category II, and Category III. Category I consists of numeric codes for established procedures with specified payment amounts or costs. This category includes six main procedure types: Evaluation and Management, Anesthesia, Surgery, Radiology, Pathology and Laboratory, and Medicine. Category II includes alphanumeric codes for supplemental purposes related to evaluation and management procedures, without specific payment values, and encompasses 11 sections, such as Composite Measures and Physical Examination. Category III codes are used for emerging technologies, temporary procedures, and services.

Similarly, the International Classification of Diseases, 10th Revision, Procedure Coding System (ICD-10-PCS) was developed as part of ICD-10 for coding inpatient medical services and procedures. ICD-10-PCS codes can be up to seven digits long, with each digit representing specific aspects such as the medical practice section, body system, and body part. However, not all reimbursement entities adhere to the ICD-10-PCS coding for inpatient procedures. Instead, some compile comprehensive CPT lists for outpatient procedures and manually aggregate inpatient procedures into a single list within their EHR systems. This approach is reflected in the dataset examined in this paper.

MEDICAL INSURANCE CLAIMS PROCESS

Medical insurance claims are categorized into two types: inpatient and outpatient. The process for handling inpatient claims starts when a patient either arrives at the hospital due to an emergency or is referred for in-hospital treatment by a specialized physician. In both scenarios, a comprehensive medical report from a specialist must be submitted for the insurance company's approval. This approval or denial is determined by medical staff or representatives from the insurance company. The supporting medical reports, which may be scanned copies of handwritten or computer-generated originals, are registered in the insurance company's system by approval staff. They enter general approval information, including the relevant ICD code, a summary description of the case based on the report, the doctor's name, the hospital's name, the admission date, and the expected cost. A challenge arises because the full scope of procedures is often not known at the time of entry but is determined after the patient's discharge. Additionally, approvals may be partial, excluding certain uncovered treatments or medications based on the insurance contract's terms and conditions.

For outpatient claims, the process is somewhat similar but with some distinctions. Preprinted physical treatment forms, each with a unique identifier called a "Form Number," are distributed to insured members. When a patient visits a doctor's clinic, the diagnosis and required treatments are handwritten on these forms. The patient then visits various service providers – such as laboratories, radiology centers, pharmacies, or physiotherapy centers – for the necessary tests, medications, or sessions. Each provider requests approval for their services, receiving a unique approval number over the phone, known as outpatient approval. Throughout the treatment cycle, the same form is used by different providers, who retain their copies and eventually compile them to submit to the insurance company for payment. This means a single medical case can result in multiple claims to the insurance company, but the unique form number links all these claims to one patient's case.

On a regular basis, typically monthly, hospitals provide hard copy files of discharged inpatient treatment details and invoices to insurance companies. These files are submitted in batches corresponding to different insured members. Similarly, outpatient service providers also submit their claims in batch form. Upon receipt, these batches are systematically numbered and registered, including details such as the provider's name, the total number of claims, and the total claimed amount. The batches are then distributed to medical insurance claims entry teams, which are usually organized by provider specialties. The claims entry staff processes each batch by registering every claim and its corresponding invoice separately. Throughout this process, both manual and systematic audits are conducted to ensure that only legitimate services and treatments are paid. Any amounts for excluded treatments are flagged and either not paid to the provider or recovered from the insured members if payment was made initially.

During the claims and invoice entry process, pre-authorized approvals are verified and linked to their respective claims. Some insurance companies' internal policies permit certain claims, such as professional fees for physical and biological examinations in doctors' clinics and pharmacy prescriptions below specific monetary thresholds, to be accepted without prior approvals.

From a data perspective, challenges arise due to the heterogeneous nature of data distributed across various EHR systems, which often suffer from data quality issues, particularly in the Jordanian healthcare sector. These challenges impact the effectiveness of robust predictive models and machine-learning applications in the medical domain. The following lists some of these challenges:

- Heterogeneous data, such as missing features, incorrect values, handwritten text, abstracted descriptions, and outliers, are problematic for any machine-learning algorithm.
- Not all relevant data is shared between the different provider types and the insurance companies. Some important features with strong influential effects on predictive models' performance, like lab results, are not always shared with insurance companies.
- No full medical data records exist in one single system as patients may visit different providers, and some of their visits are at their expense, not on the insurance company side, which makes some important historical data missing in insurance systems.
- A lot of noisy data exists due to wrong data entry practices. For example, diagnoses are not registered at the correct level of detail, as mentioned earlier. Treatment codes are not entered correctly for many medical cases.
- The ICD and treatment lists contain a huge number of instances that limit the ability to use well-known preprocessing techniques, such as one-hot encoding, as these create sparse data representations. Accordingly, some specialized data representation techniques may be required to cater to this problem.
- Due to the high volume of medical claims and the high number of daily claims targets imposed on insurance medical claims staff members, some details are skipped during the claims entry process. For example, medicines and laboratory tests per clinical case are not entered in detail, but rather, information on medicines is entered under single high-level treatment code categories such as Local Medicines and Foreign Medicines, and all laboratory tests are aggregated under the LABS CPT category.

In our medical dataset, many challenges exist that prevent us from utilizing some important features due to the high empty value rates, such as patient-specific features, including age, gender, and Body Mass Index (BMI).

RELATED WORK

In this paper, we conducted a literature review to identify relevant works and potential approaches for addressing the problem of medicine ranking (Li et al., 2022; Miyachi et al., 2023; Torfi et al., 2022; Zeng et al., 2022; Zhang et al., 2022). Most related literature primarily addresses disease classification, specifically ICD coding clinical cases. These studies focus on predicting the relevant ICD codes for clinical cases by applying learning algorithms to heterogeneous clinical texts associated with these cases. Fewer studies target the prediction of likely future diseases based on patient profiles and historical medical data, with many of these focusing on the timing of disease occurrence. Very few studies, however, address the prediction of treatments based on patients' historical data.

The relevant literature can be categorized into two main areas: medical treatment prediction and medicine prediction and ranking. Research on medical treatment prediction does not specifically focus on medicines but rather on a range of treatment types, including procedures, lab tests, and medicines. The second category focuses directly on medicines using two approaches: prediction and ranking.

In the realm of medical treatment prediction, only a few studies address this problem, with all treating it as a classification issue (Kelly, 2019; Zeng et al., 2022; Zhang et al., 2022). These studies aim to predict procedure codes for treatments using features related to patients, diseases, and clinical texts. Subotin and Davis (2014) proposed a system for predicting treatment codes based on clinical text features recorded in EHR systems. Their system leverages the hierarchical structure of ICD-10-PCS codes, categorizing them into two levels: a high-level concept and a lower-level code. High-level concepts are extracted and mapped to the ICD-10-PCS code's first and second digits, representing the Section and Body System, respectively. Regularized logistic regression classifiers are then applied to obtain confidence scores for medical concepts within the EHR narratives. Based on these scores, confidence scoring is calculated for codes under the predicted concept hierarchies. The authors used a modified version of the Mean Reciprocal Rank (MRR) metric to evaluate performance, achieving the best results with an MRR of 0.572.

Levy et al. (2022) compared the performance of three text classification algorithms – XGBoost, Support Vector Machine (SVM), and BERT – using a dataset of pathological reports to predict treatment codes associated with these reports' narratives. As a dimensionality reduction step, the authors utilized topic modeling techniques, specifically UMAP and LDA, to transform the data into a format suitable for prediction. XGBoost and BERT demonstrated comparable performance, achieving median AUCs of 0.997 and 0.995, respectively. Haq et al. (2017) proposed a deep learning approach for treatment coding using an artificial neural network (ANN). This method employed separate embedding matrices for each digit of medical claims ICD codes, which were then concatenated to form a dense representation of the complete ICD code. The proposed model outperformed a probabilistic-based model and the association rule mining algorithm, apriori, achieving a Recall@3 of 90% and a Precision@3 of 45%.

In the domain of medicine prediction and ranking, most studies have focused on either predicting medicines or ranking them based on efficacy and toxicity rather than prescription frequencies, which is the focus of our proposed approach (Jin & Garg, 2023; Lu, 2023; Ru et al., 2022; Vaishya & Misra, 2022; S. Yang et al., 2021). To the best of our knowledge, no prior work has addressed the problem of LTR for medicines based on prescription frequencies.

Kumar et al. (2021) introduced a multi-layer neural network approach for predicting general medicines by identifying accurate medical combinations for patients who cannot find suitable medications at pharmacies. The proposed methodology utilized Quantum Neural Networks (QNN) to predict appropriate general medicines based on disease symptoms. This model achieved 95% accuracy, surpassing SVM, Random Forest, and Naïve Bayes classifiers.

Recent advancements in machine learning, deep learning, and LTR algorithms have been applied to predict and rank drugs. Several studies have illustrated the use of these algorithms to make predictions and rankings based on factors such as efficacy, toxicity, and side effects (Gerdes et al., 2021).

In the field of medicine prediction, Lavecchia (2019) introduced a deep-learning approach to predict drug efficacy and toxicity. The study employed a public chemical biology dataset to evaluate their model, utilizing deep learning algorithms such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs). The model achieved a root mean square error (RMSE) of 0.32 for drug efficacy prediction and 0.29 for drug toxicity prediction. Similarly, Zhu et al. (2021) applied machine learning algorithms to predict drug efficacy using genomic data. Their model was evaluated using publicly available genomic datasets that incorporated algorithms, including support vector machines (SVMs) and random forest (RF). The reported RMSE for drug efficacy prediction was 0.32, and for drug toxicity prediction, it was 0.29.

In the domain of medicine ranking, Gerdes et al. (2021) employed deep learning and network analysis to rank drugs based on efficacy and toxicity. Using a public chemical biology dataset, their model integrated deep learning algorithms with network analysis techniques, achieving a Mean Reciprocal Rank (MRR) of 0.76. Zhou et al. (2020) proposed a machine-learning approach for ranking drugs based on efficacy and side effects. Their model, which was evaluated using a publicly available chemical biology dataset, utilized algorithms such as SVMs and RF. This approach achieved a mean average precision (MAP) of 0.82 for drug ranking based on efficacy and side effects. Chakradhar (2017) utilized LTR algorithms to rank drugs for repositioning, leveraging heterogeneous data sources. The study integrated public chemical biology datasets with drug-disease association data to evaluate their model. The LTR algorithms demonstrated effectiveness, achieving a mean reciprocal rank (MRR) of 0.73 for drug repositioning based on these diverse data sources.

Thus, Table 1 presents the key studies in the literature on medicine ranking. The findings indicate that machine learning, deep learning, and LTR algorithms hold significant promise for improving the prediction and ranking of medicines. These algorithms can enhance the accuracy and efficiency of predictions and rankings, making them valuable tools in the medical field. The limited use of LTR approaches in existing studies further underscores the relevance and motivation for our work.

Table 1. Related works summary

Reference	Dataset	Approach	Training testing split	Algorithms	Measure	Performance
Subotin and Davis (2014)	Corpus of 28,536 EHRs (individual clinical records)	Calculating scores for the first and second digits of CPT generating concepts for them and estimating their probability	5-fold cross-validation	Logistic regression	Modified version of Mean Reciprocal Rank (MRR)	MRR = 0.572
Haq et al. (2017)	2.3 million insurance claims from a U.S.-based billing company	Embedding layer to handle one-hot encoded matrix for the 7 digits of ICD 9 code	2.3 million claims for training and 70k claims for testing	ANN apriori	Precision Recall@k	Recall @ 3 = 90% Precision @ 3 = 45%
Levy et al. (2022)	93,039 pathology reports	Topic modeling techniques using UMAP and LDA	5-fold cross validation	XGBoost SVM BERT	AUC	XGBoost AUC = 0.997 SVM AUC = 0.977 BERT = 0.995
Kumar et al. (2021)	Data from private EHR System	Deep learning models called quantum neural networks are used to predict general medicines.	Not mentioned	ANN	Accuracy	SVM: 85%; RF: 88%; NVB: 90%; QNN: 95%
Li et al. (2022)	Publicly available chemical biology dataset	Deep learning algorithms and network analysis to rank drugs based on their efficacy and toxicity	5-fold cross validation	ANN	Mean Reciprocal Rank (MRR)	MRR = 0.76
S. Yang et al. (2021)	Publicly available chemical biology	Machine learning algorithms, including support vector machines (SVMs) and random forest (RF)	70% training, 20% validation and 10% testing	SVM RF	Mean Average Precision (MAP)	MAP = 0.82

METHODOLOGY

In this section, we explain the proposed model and the steps required to build the model, such as data preparation and preprocessing. Moreover, we will take a deep look at the available LTR models and the relevant LTR evaluation metrics.

THE DATASET

The datasets used in this study include confidential information on medical treatments sourced from an insurance company. To ensure privacy, the dataset has been anonymized, with personally identifiable information (PII) removed. Each insured member and healthcare provider has been assigned a unique identification number. The dataset is organized into monthly cohorts of patient treatments, with the same disease potentially appearing multiple times and accompanied by varying lists of medications, often derived from different active ingredients. Treatment records from different months have been designated as test datasets, while records from other months have been used for training and validation purposes. The raw dataset contains historical treatment records that demonstrate the co-occurrence of diseases and medications based on the frequency of patient visits to healthcare providers for the same condition and the treatments received. This historical treatment data includes 21 features, as detailed in Table 2, and Table 3 presents key statistics related to the raw data.

Table 2. Raw dataset features description

Feature no.	Feature name	Data type	Description	Data sample
1	Case no.	Numerical	The unique medical case number	11595749
2	Patient ID	Numerical	The patient's unique identifier	149618
3	Claim ID	Numerical	The unique claim identifier of the patient's medical case	14111353
4	Treatment date	Date	The date of medical case occurrence	27/12/2021
5	Provider ID	Numerical	The medical provider's unique identifier	565720
6	ICD code	Categorical	The ICD code for the medical case diagnosis	J00
7	ICD name EN	Categorical	The ICD name in English for the medical case diagnosis	Acute nasopharyngitis (common cold)
8	ICD name AR	Categorical	The ICD name in Arabic for the medical case diagnosis	الرشح
9	Main parent code	Categorical	The main ICD code for which the current disease belongs	J00–J99
10	All names to the main parent	Categorical	The list of all disease names in the tree branch from the current disease code to the main parent code	Acute nasopharyngitis (common cold) || --> || acute upper respiratory infections || --> || diseases of the respiratory system
11	Service ID	Numerical	Unique Identifier of Medical Treatment (medicine)	74425
12	Service code	Categorical	Unique code of medical treatment (medicine)	PHR1031
13	Medicine name	Categorical	The name of the medicine	Biodal 5000 IU tab (60)
14	Generic name	Categorical	The generic name of the medicine	(Cholecalciferol (vitamin D3): 5000 IU) cap/tab [oral]
15	Unit type	Categorical	The unit in which the medicine is sold	Tablet

Feature no.	Feature name	Data type	Description	Data sample
16	Dosage form	Categorical	The form in which the medicine is given to the patients	Cap/Tab
17	Package type	Categorical	The package type in which the group of medicine unit types are packed and distributed	Container
18	Package size	Categorical	The smallest size in which the medicine packages are distributed	60s
19	Ingredients	Categorical	The active ingredient of the medicine	Cholecalciferol (vitamin D3)
20	Strength	Categorical	The strength of the active ingredient per each size unit of the medicine	5000 IU
21	Dosage unit	Categorical	The unit in which the medicine's dosage is sold	Tablet

THE PROPOSED APPROACH

Our approach involves developing pointwise deep-ranking models to first predict the probabilities of active ingredients in medicines for each disease. Subsequently, for the top *k* predicted active ingredients, we employ another pointwise deep-ranking model to estimate the probabilities of medicines associated with each disease and its corresponding ingredients. This two-stage process aims to accurately predict the top medicine for each active ingredient used in treating each disease. We tested two values for *k* – the average and the median number of active ingredients per disease – as these are commonly used configurations.

To evaluate the robustness of our approach and models, we conducted 18 cross-validation trials, training prediction models and assessing their performance on different testing datasets. This number of trials was chosen because our primary dataset spans treatments over 18 different months. For each trial, we used one month of treatment data as the test dataset, another month for validation, and the remaining 16 months for training. In each trial, we developed a separate deep-learning regression model for each label to predict its probability values. This resulted in 201 deep-learning regression models for the ingredient prediction task across the 18 trials. For the medicine prediction task, the number of models varied according to the number of unique medicines in each training dataset, with a maximum of 1,754 unique medicines, as indicated in Table 3. To create the necessary ranking models, the datasets for each trial were prepared to ensure that ingredient and medicine probabilities could be accurately calculated and predicted based on features relevant to the diseases.

Table 3. Important dataset statistics

Statistic	Value
Number of instances	53,792
Number of cases (visits)	18,204
Number of patients	13,547
Number of unique diseases	237
Number of unique ingredients	201
Number of unique medicines	1,754
Average number of unique ingredients per disease	8
Median number of ingredients per disease	3
Maximum number of unique ingredients per disease	99
Average number of unique medicines per disease	46
Maximum number of unique medicines per disease	603
Average number of unique medicines per disease and ingredient	6
Maximum number of unique medicines per disease and ingredient	33

The dataset statistics presented in Table 3 indicate that, on average, each disease is associated with medicines from approximately eight unique ingredients, while the median number of ingredients per disease is 3. These figures correspond to the *k*-predicted ingredients in the first model of ingredient prediction, as previously mentioned. Figure 2 illustrates the implementation steps of the proposed approach. To evaluate the effectiveness of our proposed method, we compared it with an alternative approach based on a well-established data representation method used in document retrieval problems. This comparison involved representing the data using the LibSVM format (Chang & Lin, 2011), where each instance of the query-document pair – disease-medicine in this context – is encapsulated with the query ID and relevance score in a standardized format.

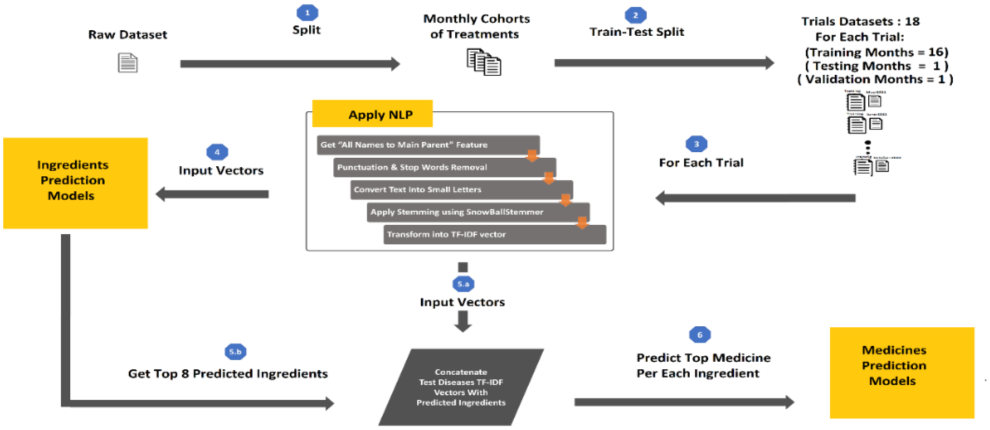
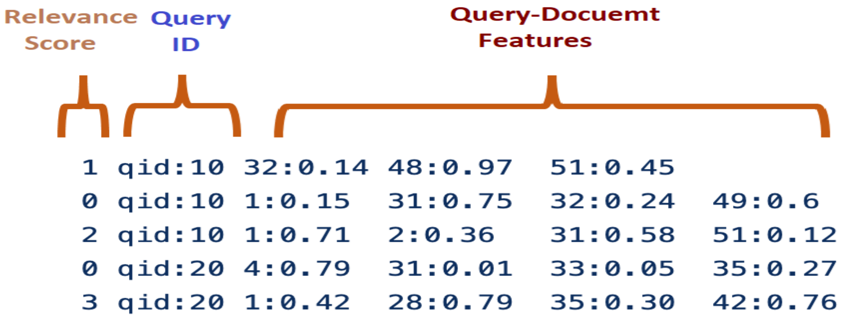

Raw Dataset
1
Split
Monthly Cohorts of Treatments
2
Train-Test Split
Trials Datasets : 18
For Each Trial:
(Training Months = 16)
(Testing Months = 1)
(Validation Months = 1)
Apply NLP
Get "All Names to Main Parent" Feature
Punctuation & Stop Words Removal
Convert Text into Small Letters
Apply Stemming using SnowBallStemmer
Transform into TF-IDF vector
3
For Each Trial
4
Input Vectors
Ingredients Prediction Models
5.a
Input Vectors
5.b
Get Top 8 Predicted Ingredients
Concatenate Test Diseases TF-IDF Vectors With Predicted Ingredients
6
Predict Top Medicine Per Each Ingredient
Medicines Prediction Models

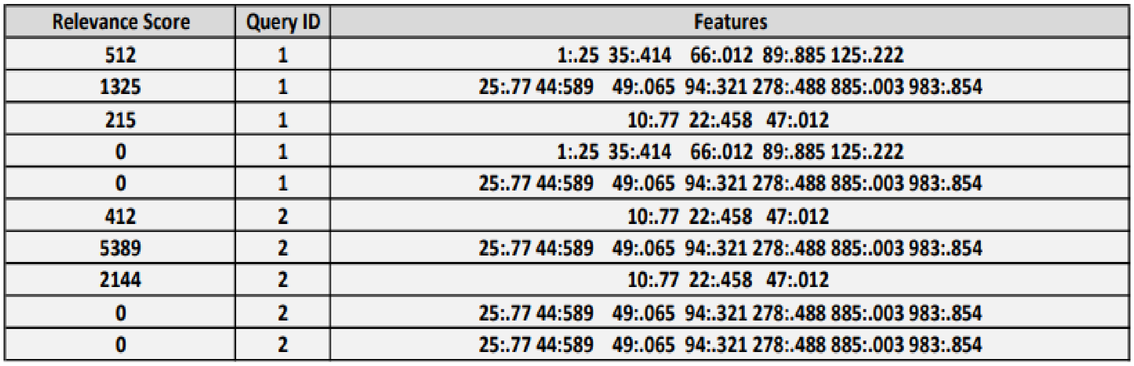
Figure 1. Proposed approach implementation steps

Chang and Lin (2011) provided an overview of the LIBSVM library, which includes tools for formatting data in the LibSVM format. The paper offers a comprehensive explanation of this format, detailing specifications for representing both sparse and dense data. The authors highlight the benefits of using the LibSVM format for machine learning tasks, including efficient data storage and retrieval. The LibSVM format is commonly employed in LTR tasks due to its straightforward and efficient implementation of various LTR algorithms, such as those used in TensorFlow ranking models (Pasumarthi et al., 2019), with different loss functions. By comparing the performance of our proposed approach with this benchmark, we aim to demonstrate the effectiveness of our method. The LibSVM format, illustrated in Figure 3, is designed for sparse data and comprises three main components: the relevance score, the query ID, and a numerically labeled list of query-document features. This representation differs from our approach, where labels are represented as one-hot vectors of medicine probabilities rather than a single numerical feature per disease-medicine instance. Figure 4 provides a sample of the data in LibSVM format. After converting our data to this format, we utilized TensorFlow's TF Ranking library – a new tool for LTR tasks in TensorFlow – to build three models representing pointwise, pairwise, and listwise algorithms. We then evaluated these models across the same 18 trials and compared their performance to our approach.



```
1 qid:10 32:0.14 48:0.97   51:0.45
0 qid:10 1:0.15  31:0.75   32:0.24   49:0.6
2 qid:10 1:0.71  2:0.36    31:0.58   51:0.12
0 qid:20 4:0.79  31:0.01   33:0.05   35:0.27
3 qid:20 1:0.42  28:0.79   35:0.30   42:0.76
```

Figure 2. LibSVM format



Relevance Score	Query ID	Features
512	1	1:.25 35:.414 66:.012 89:.885 125:.222
1325	1	25:.77 44:589 49:.065 94:.321 278:.488 885:.003 983:.854
215	1	10:.77 22:.458 47:.012
0	1	1:.25 35:.414 66:.012 89:.885 125:.222
0	1	25:.77 44:589 49:.065 94:.321 278:.488 885:.003 983:.854
412	2	10:.77 22:.458 47:.012
5389	2	25:.77 44:589 49:.065 94:.321 278:.488 885:.003 983:.854
2144	2	10:.77 22:.458 47:.012
0	2	25:.77 44:589 49:.065 94:.321 278:.488 885:.003 983:.854
0	2	25:.77 44:589 49:.065 94:.321 278:.488 885:.003 983:.854

Figure 3. LibSVM format

DATA PREPARATION AND PREPROCESSING

Data preprocessing is a critical step that significantly impacts the performance of predictive models. As outlined in Table 2, the dataset primarily consists of categorical features, with a few numerical and date features. Given that our models will be built using deep learning techniques, extensive preprocessing is not required. Identifier features, along with certain medicine-specific attributes such as medicine name, dosage form, package type, package size, strength, and dosage unit, have been excluded. These features were removed because medicines are the target labels we aim to rank, and our data processing pipeline focuses on predicting medicine ingredients as an intermediary step toward predicting the medicines themselves. Additionally, medicine name features were omitted since their information is incorporated into a more comprehensive feature, "All Names to Main Parent."

In alignment with the proposed approach, we generated 18 distinct datasets from the original dataset. Each dataset was divided into three subsets: a test dataset comprising one month of treatments, a validation dataset containing a different month of treatments, and a training dataset with the remaining 16 months of treatments from the original dataset. To develop deep ranking models as outlined in the previous section, each training, validation, and testing dataset was aggregated by disease-medicine features. This aggregation enabled the calculation of ingredient and medicine probabilities for each split of the overall dataset across all trials. As a result of this aggregation process, the total number of instances was significantly reduced from 53,792 to 12,585. Additionally, four new features were created to represent ingredient counts and ratios, as well as medicine counts and ratios for each disease. Table 4 presents the aggregated dataset and the calculated probabilities, including three sample records to provide further insights.

Table 4. The resulting dataset after aggregation and probabilities calculation

Feature name	Feature description	Sample record 1	Sample record 2	Sample record 3
ICD code	International code of disease (ICD)	R05	J00	N39.0
Main parent code	The main ICD code for which the current disease belongs	R00–R99	J00–J99	N00–N99
All names of the main parent	The list of all disease names in the tree branch from the current disease code to the main parent code	Cough || --> || symptoms and signs involving the circulatory and respiratory systems || --> || symptoms, signs, and abnormal clinical and laboratory findings not elsewhere classified	Acute nasopharyngitis (common cold) || --> || acute upper respiratory infections. || --> || diseases of the respiratory system	Urinary tract infection, site not specified || --> || other disorders of urinary system || --> || other diseases of the urinary system || --> || diseases of the genitourinary system
Service code	Unique code of medical treatment (medicine)	PHR5066	PHR1031	PHR1045
Ingredients	The active ingredient of the medicine	Mometasone furoate	Cholecalciferol (vitamin D3)	CEFIXIME
Ingredient in ICD Count	The frequency of using the current ingredient with the current disease in the historical data for the current RECORD_TYPE	1	5	5
Ingredient in ICD Ratio	The frequency ratio of using the current ingredient with the current disease proportional to all ingredients' frequencies used to the same disease in the historical data for the current RECORD_TYPE	0.09090909	0.00740741	0.23809524
Medicine in ICD Count	The frequency of using the current treatment (medicine) with the current disease in the historical data for the current RECORD_TYPE	1	1	1
Medicine in ICD Ratio	The frequency ratio of using the current treatment (medicine) with the current disease proportional to all treatments (medicines) frequencies used for the same disease in the historical data for the current RECORD_TYPE	0.09090909	0.00148148	0.04761905

In our approach, we propose a two-step modeling process: first, for predicting ingredients, and second, for predicting medicines. This requires creating two distinct datasets from the original aggregated dataset for each trial, one for each model. The ingredients prediction dataset is constructed by utilizing only the calculated ingredient probabilities for each disease. This dataset contains one record per disease, including disease features and a list of all unique ingredients with values representing each ingredient's associated ratio or probability with the disease in the current record. Similarly, the medicines prediction dataset is created by focusing on medicine probabilities. In this dataset, the labels list consists of medicine probabilities, and the ingredient feature is included as a one-hot encoded list of all ingredients associated with each disease. The categorical features "Main Parent Code" and "ICD code" for medicines are also represented as one-hot encoded vectors. We observed that the textual feature "All Names to Main Parent" is highly relevant to our labels, as it provides a description of the diagnosed disease and its hierarchical parent diseases in the ICD classification. Therefore, this feature was included in our input data.

In handling textual data within deep learning models, natural language processing (NLP) techniques have been employed to process the "all names to main parent" textual feature. This process involved removing punctuation and stop words using the NLTK library, converting all words to lowercase, applying stemming with SnowballStemmer, and representing the feature as a TF-IDF vector using scikit-learn's TfidfVectorizer. Table 5 illustrates a sample of the "all names to main parent" feature before and after NLP processing while Table 6 presents a sample of the resulting text after TF-IDF vectorization. For the benchmark approach, each instance of the second dataset was formulated by assigning medicine probabilities as score values and providing a unique query ID for each disease code. The feature vectors were formatted to comply with the LIBSVM specification, where feature vector values are preceded by a feature number and a colon.

Table 5. Sample of "all names to main parent" feature before and after NLP
(The output of the NLP tokenization process is partial words.)

All names to main feature - before NLP	After NLP
Cough || --> || symptoms and signs involving the circulatory and respiratory systems || --> || symptoms, signs, and abnormal clinical and laboratory findings not elsewhere classified	Cough symptom sign involv circulatori respiratori system symptom sign abnorm clinic laboratori find elsewher classifi
Pain in throat and chest || --> || symptoms and signs involving the circulatory and respiratory systems || --> || symptoms, signs, and abnormal clinical and laboratory findings not elsewhere classified	Pain throat chest symptom sign involv circulatori respiratori system symptom sign abnorm clinic laboratori find elsewher classifi
Abdominal and pelvic pain || --> || symptoms and signs involving the digestive system and abdomen || --> || symptoms, signs, and abnormal clinical and laboratory findings, not elsewhere classified	Abdomin pelvic pain symptom sign involv digest system abdomen symptom sign abnorm clinic laboratori find elsewher classifi
Nausea and vomiting || --> || symptoms and signs involving the digestive system and abdomen || --> || symptoms, signs, and abnormal clinical and laboratory findings not elsewhere classified	Nausea vomit symptom sign involv digest system abdomen symptom sign abnorm clinic laboratori find elsewher classifi
Symptoms and signs involving the nervous and musculoskeletal system || --> || symptoms, signs, and abnormal clinical and laboratory findings not elsewhere classified	Symptom sign involv nervous musculoskelet system symptom sign abnorm clinic laboratori find elsewher classifi

Table 6. All names to main parent feature after TDF-IDF vectorization

Item	Value
Processed text	abdomin pain abdomin pelvic pain symptom sign involv digest system abdomen symptom sign abnorm clinic laboratori find elsewher classifi
Shape of vectorized version	(1, 2464)
Vectorized version	array ([0.14585238, 0.14585238, 0.14585238, ..., 0. , 0. , 0.])

LEARN TO RANK (LTR) APPROACHES

LTR approaches originated in the late twentieth century and gained prominence with the widespread adoption of the internet and the World Wide Web, particularly within search engines, which serve as the primary interface for content searchers. The demand for LTR arose specifically to address information retrieval (IR) problems, where the objective is to deliver the most relevant documents in response to user queries, ordered to reflect their relevance, with the most pertinent documents appearing at the top of the list.

LTR is considered a supervised machine learning technique wherein ranking models are generated automatically based on labeled training datasets (Shi et al., 2010). Given that the primary goal of LTR algorithms is to rank results by relevance or importance, specialized metrics have been developed to evaluate these models. Common metrics in this domain include mean average precision (MAP), mean reciprocal rank (MRR), Precision@n, Recall@n, discounted cumulative gain (DCG), and normalized discounted cumulative gain (NDCG). LTR approaches can be categorized into three main types: pointwise, pairwise, and listwise. Pointwise methods assign numerical or ordinal ranking scores to individual items based on a loss function, subsequently ordering all documents according to these scores. These methods are typically treated as regression problems because the predicted results are numerical scores. Pairwise approaches, in contrast, evaluate pairs of documents and rank them according to the differences in their relative scores. The loss function in pairwise methods focuses on the relevance difference between pairs. A notable limitation of pairwise approaches is their computational intensity, which can restrict scalability and application to large datasets (Shi et al., 2010). Prominent pairwise LTR algorithms include SVMRank (Joachims, 2002), RankBoost (Freund et al., 2003), RankNet (Burges et al., 2005), and LambdaRank (Burges et al., 2010). Listwise approaches, on the other hand, use entire lists of items or instances for learning. They are distinguished from other LTR methods by their objective of directly optimizing ranking metrics. Listwise approaches have demonstrated superior performance compared to pairwise methods. Examples of popular listwise LTR algorithms include LambdaMart and ListNet (Cao et al., 2007).

Although LTR approaches have predominantly focused on information retrieval (IR) problems, their applicability and popularity have also extended to recommender systems. In the context of recommender systems, the ranked results pertain to tangible and intangible items, products, and services such as movies, books, and e-commerce goods. Two well-known techniques in recommender systems are content-based filtering and collaborative filtering (CF). Content-based filtering relies on data features related to the items themselves, while CF, recognized as one of the most successful recommendation techniques (Shi et al., 2010), leverages user-profiles and behavioral data.

In recommender systems, the analogy to IR is that item or user behavior features represent queries, and the resulting items correspond to the retrieved documents. Various methods exist for applying LTR approaches in recommender systems. For collaborative filtering, examples of listwise methods include CLiMF (Shi et al., 2012a), CoFiRank (Weimer et al., 2007), and TFMAP (Shi et al., 2012b). Pairwise methods include EignRank, Probabilistic Latent Preference Analysis (Liu & Yang, 2008; Liu et al., 2009), and Bayesian Personalized Ranking (BPR) (Rendle et al., 2012). Listwise approaches generally offer better scalability.

As described in the methodology section, we evaluate our proposed approach by comparing it with a benchmark approach that employs three LTR algorithms – pointwise, pairwise, and listwise. Specifically, we apply RankNet and ListNet algorithms to our data, which has been represented in LibSVM format and processed using the TensorFlow Ranking library.

Evaluation Metrics

Several metrics are available for evaluating the performance of ranking algorithms. Some metrics are designed for binary relevance ranking problems, where the focus is on predicting relevant documents and positioning them at the top of the prediction list without considering their order among each other. In such cases, if two algorithms successfully place all *k* relevant results within the top *k* predicted list, they receive the same score of "1," regardless of the specific order of the items within these top *k* results.

Recall@k and Precision@k are examples of binary classification metrics adapted to measure the performance of LTR algorithms. The variable *k* represents the number of top items of interest. For instance, if we are interested in the top ten recommended results, *k* is set to 10. Recall@k measures the proportion of correctly identified relevant items within the top *k* predicted results compared to the total number of relevant items or documents. Conversely, Precision@k assesses the proportion of correctly identified relevant items within the top *k* predicted results relative to the total number of retrieved items or documents. Precision@k is defined as follows:

$$Precision@k = \frac{\#\ of\ Relevant\ Items\ in\ the\ top\ k\ retrieved\ results}{\#\ of\ Retreived\ Items} \tag{1}$$

$$Recall@k = \frac{\#\ of\ Relevant\ Items\ in\ the\ top\ k\ retrieved\ results}{\#\ of\ Relevant\ Items} \tag{2}$$

where *k* is the number of top items or documents we are interested in.

Mean average precision (MAP) is an appropriate metric for our diagnosis prediction problem because it accounts for the ranks of results and provides a comprehensive numerical measure of overall classifier or ranker performance. Average precision (AP) is computed by averaging the precision values at the ranks of relevant items or documents while excluding precision values at irrelevant ranks. This metric is weighted towards the top of the ranking, meaning that a higher rank (e.g., rank 1) has more influence on the score than lower ranks (e.g., rank 2, rank 3, etc.). AP can be calculated for any number *n* of top-ranked items, where *n* represents the total number of predicted labels. The AP is expressed as follows:

$$AP_n = \frac{1}{\#\ of\ Relevant\ Items}\sum_{k}^{n} Precision@k \text{ x } rel@k \tag{3}$$

where *n* refers to the total number of documents that we are interested in, and rel@k is a relevance function that equals 1 if the item, or document, at rank *k* is relevant and equals 0 otherwise.

The mean average precision is the mean of total instances or queries' average precisions. The mean average precision is expressed as the following:

$$mAP = \frac{1}{n}\sum_{i=1}^{N} \mathrm{AP}_i \tag{4}$$

where *n* is the number of instances or queries.

As indicated in Equation (3), average precision (AP) is computed for binary relevance classification, where each class label is considered either relevant or irrelevant. However, in some scenarios, measures of relevance are more nuanced, such that the relevance of an item is assessed with a score rather than a simple binary classification. In such cases, both the presence of relevant items within the top *k* results and their order based on relevance scores are crucial. For instance, if all five relevant items appear within the top five ranks, the AP would be 1. Nevertheless, in document retrieval tasks, this does not fully capture the desired outcome. The relevance scores of the documents significantly impact their ranking, and the goal is to prioritize the most relevant documents at the top, followed by the next most relevant, and so on.

To address the need for ranking relevant labels more comprehensively, metrics have been developed that not only account for the retrieval of relevant labels but also consider their order based on predefined relevance scores. These metrics ensure that labels with higher relevance scores contribute more to the overall metric value if they appear earlier in the prediction list rather than later.

A prominent example of such a metric is normalized discounted cumulative gain (NDCG). NDCG takes into account the order of top-ranked relevant documents and penalizes irrelevant items at the top of the list more heavily than those at the bottom. NDCG@k, where *k* represents the number of top-ranked items of interest, is a common formulation. Discounted cumulative gain (DCG) measures relevance for a single query but does not provide a comprehensive measure for an entire ranking algorithm due to the variability in query results. NDCG addresses this issue by normalizing the DCG across all queries and dividing each query's DCG by the DCG of the ideal ranking or ground truth ranked results. The DCG is expressed as follows:

$$DCG@_k = \sum_{i=1}^{k} \frac{rel_i}{log_2(i+1)} \tag{5}$$

where rel_i, is the graded relevance of the result at position i, and k is the top-rank position. nDCG is expressed as the following:

$$nDCG_k = \sum_{i=1}^{k} \frac{DCG_k}{IDCG_k} \tag{6}$$

where $IDCG_k$ is the ideal DCG for the top-rank position *k*.

The NDCG metric has certain limitations. One notable issue with DCG is that it does not penalize missing or erroneous data in the results, allowing systems that produce irrelevant or incorrect results to still achieve high scores. This can obscure the true quality of the system's output. Additionally, DCG may not accurately assess performance in cases where multiple results are equally relevant. In such situations, the score might not reflect the actual quality of the results. Given these limitations, we believe NDCG is suitable for our needs in ranking and ordering medical treatments, as it aligns with our goal of listing the top relevant medicines in a manner that accurately represents a doctor's preferences for the medical case. Consequently, we have chosen to use NDCG@k as our performance metric.

IMPLEMENTATION

This section describes the implementation of the prediction models in detail, including our data process pipeline pointwise deep ranking models and the benchmark approach models. Moreover, the results are presented and discussed thoroughly.

Implementation Environment

The models were developed using Google Colab, an online platform provided by Google that facilitates the execution of Python code and the utilization of various libraries. During the implementation phase, several data science-related Python libraries were employed, including NumPy, Pandas, NLTK, Matplotlib, Scikit-learn, TensorFlow, and TensorFlow Ranking.

Building the First Model (Ingredients Prediction)

The initial model was constructed with a single input layer designed to accommodate a float vector of size 2,723, representing the TF-IDF vectorized textual data from the “ICD All Names to Main Parent” feature. This feature was generated during data preprocessing, following the concatenation with one-hot encoded vectors corresponding to the “Main Parent Code” and “ICD Code” features. Separate hidden layers were implemented for each of the 201 unique ingredient output labels within the model. To address the risk of overfitting, dropout layers were incorporated after the first and second hidden layers, each with a dropout rate of 20%. Each branch of hidden and dropout layers led to a single output layer, with one output node predicting probability values for the specific label.

For each label prediction, input data vectors were processed through a TensorFlow Keras embedding layer, with an input dimension of 10,000 and an output dimension of 128. The embedding layer was initialized uniformly and had an input length of 25, reflecting the maximum word length of the “All Names to Main Parent” feature in the training dataset after applying natural language processing techniques. The output tensor from the embedding layer was directed to a GlobalAveragePooling1D layer, which averaged the values across all input vectors for each of the 128 dimensions, producing a tensor of shape (None, 128).

This tensor was then passed through two hidden dense layers, both employing the “ReLU” activation function. The first hidden dense layer contained 32 nodes, followed by a second layer with eight nodes. The tensor emerging from the dropout layer after the second hidden layer was finally fed into a dense output layer, utilizing a “Linear” activation function and featuring a single node, as previously described. Figure 5 illustrates the components of the first model.

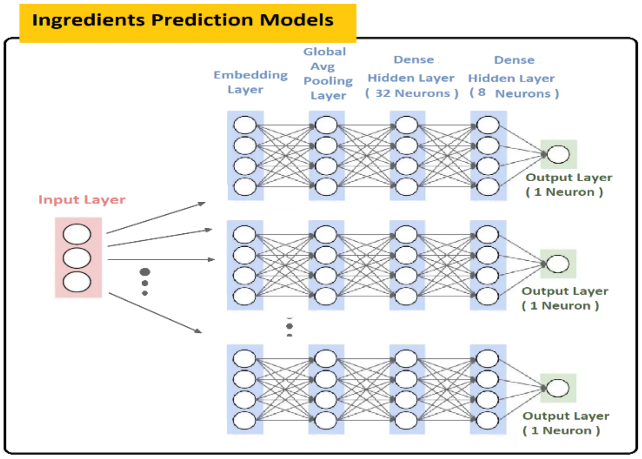

Ingredients Prediction Models
Embedding Layer
Global Avg Pooling Layer
Dense Hidden Layer (32 Neurons)
Dense Hidden Layer (8 Neurons)
Input Layer
Output Layer (1 Neuron)
Output Layer (1 Neuron)
Output Layer (1 Neuron)

Figure 4. First model – Ingredients Prediction Models

The models were compiled using the mean squared error (MSE) loss function, chosen due to the sparse nature of the ingredient labels vector, which contains numerous zero probability values alongside labels with very low probabilities relative to others with higher probabilities. MSE was particularly effective in penalizing the misprediction of extreme values. The ADAM optimizer was employed, and it is known for its efficiency in stochastic optimization, particularly in regression tasks. A learning rate of 0.01 was selected, which produced favorable results after experimenting with various rates. Finally, the model was trained on the training dataset and validated on the validation dataset across 100 epochs. The number of epochs was determined based on the maximum required by one of the top 10 occurring ingredient labels' loss curves. Notably, for the ingredient label "DESLORATADINE," the model showed stabilization in loss reduction around the 100th epoch, as depicted in Figure 6.

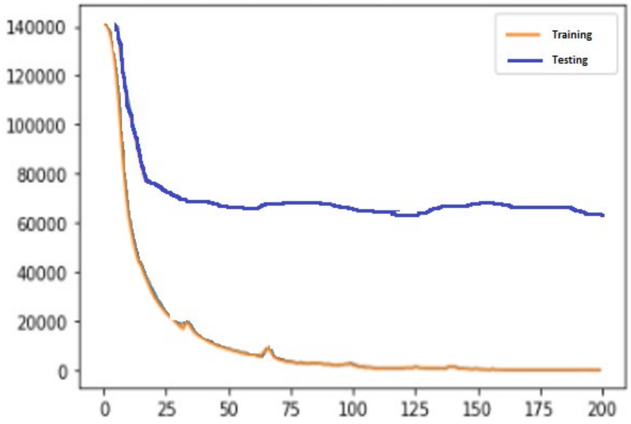


Figure 5. Loss curves for the "DESLORATADINE" ingredient label

Building the Second Models (Medicines Prediction)

For the second set of medicine prediction models, the same training and testing datasets were utilized. However, these models incorporated the medicine ingredients as part of the input vector. New models were created based on the same diseases and all associated ingredients from the training dataset. Similar to the initial ingredient prediction models, the input data were used in a single input layer, after which separate model branches were developed for each output label. In this case, the output labels represent the predicted probabilities of medicines for each disease. These models were subsequently employed to predict the medicine probabilities for each disease in the test dataset after associating each disease with its top eight predicted ingredients from the first set of models. For each disease-ingredient combination, the medicine with the highest predicted probability among all relevant medicines was selected, resulting in eight predicted medicines per disease.

The input vector from the initial ingredient prediction model is concatenated with a one-hot encoded vector representing the medicine ingredients, as previously described. For each label prediction path, similar to the ingredient prediction models, the input vectors are processed through an embedding layer, followed by global average pooling, two hidden layers, each accompanied by a dropout layer, and a final dense output layer. All layers utilize the same activation functions and neuron counts as in the first models. However, the number of training epochs is reduced to 30, determined after analyzing the loss curves of the top ten medicines, as was done for the initial model.

Implementing the Benchmark Approach's Models

As detailed in the proposed approach subsection, our objective is to compare our proposed approach with an alternative approach where data is represented in LIBSVM format. In this comparison, we develop three deep learning models corresponding to pointwise, pairwise, and listwise ranking algorithms. After converting the data into LIBSVM format, the features are inputted in padded batches of 32. These input batches undergo batch normalization. Subsequently, we constructed three models with an input layer shaped as batch size and the number of features. Each model includes three hidden layers, consisting of 128, 64, and 32 neurons, respectively, each employing the ReLU activation function. The output layer of each model contains a single neuron with a linear activation function. The models are compiled using the Adam optimizer with a learning rate of 0.01. For the pointwise model, we applied the Mean Squared Error loss function, while the pairwise model utilized Pairwise Hinge Loss (Goodfellow et al., 2016), and the listwise model used the List MLE loss function (Jansen et al., 2008). The following is a description of each loss function used:

- **Mean Squared Error (MSE)**: MSE is a commonly used loss function for regression problems. In TensorFlow ranking, MSE is used to measure the difference between predicted scores and ground truth labels. The loss is calculated by squaring the difference between the predicted scores and the ground truth labels and averaging the squared differences over all examples. The goal is to minimize the MSE to improve the accuracy of the predictions.
- **Pairwise Hinge Loss**: Pairwise Hinge Loss is a commonly used loss function for ranking problems. TensorFlow ranking is used to optimize the ranking order of items. The loss function measures the difference between the predicted scores of two items and a margin value. If the difference between the scores of two items is larger than the margin value, the loss is 0. Otherwise, the loss is the difference between the scores and the margin value. The goal is to minimize the pairwise hinge loss to improve the ranking order of items.

Listwise maximum likelihood estimation (ListMLE) is a loss function for listwise ranking problems. In TensorFlow ranking, it is used to optimize the overall ranking order of a list of items. The loss function is based on the Maximum Likelihood Estimation (MLE) principle and measures the likelihood of observing the ground truth ranking order. The goal is to maximize the likelihood of observing the ground truth ranking in order to improve the overall ranking performance.

RESULTS

After conducting 18 trials on historical medical treatments, we calculated the NDCG@8 and NDCG@3 scores for both the ingredient prediction and medicine prediction models. As outlined in the approach section, the k values of 8 and 3 were chosen to represent the average and median number of ingredients per medicine, respectively. Table 7 presents the NDCG@3 results for each trial and a summary of all trials, while Table 8 provides the NDCG@8 results for the same trials.

Figure 7 illustrates a comparative analysis of the NDCG@8 versus NDCG@3 results across all models. The results show consistency between NDCG@3 and NDCG@8 scores across the 18 trials, with NDCG@8 scores consistently higher than NDCG@3 scores. This is expected, as an increase in k allows the algorithm to consider more items in the ranking, potentially leading to a higher score if the additional items are relevant to the user. However, if the additional items are irrelevant, the score remains unchanged.

Table 7. The results (NDCG@3)

Trial no.	Test month	First models (ingredient prediction)	Second models (medicines prediction)	Benchmark model (pointwise) mean squared loss	Benchmark model (pairwise) pairwise hinge loss	Benchmark model (listwise) list MLE loss
1	May-21	88	73	71	72	75
2	Jun-21	85	72	69	69	71
3	Jul-21	85	75	74	74	74
4	Aug-21	88	69	69	69	70
5	Sep-21	83	65	64	66	66
6	Oct-21	86	69	71	73	70
7	Nov-21	85	74	71	74	73
8	Dec-21	86	76	74	76	76
9	Jan-22	80	65	66	65	63

Trial no.	Test month	First models (ingredient prediction)	Second models (medicines prediction)	Benchmark model (pointwise) mean squared loss	Benchmark model (pairwise) pairwise hinge loss	Benchmark model (listwise) list MLE loss
10	Feb-22	89	75	68	67	62
11	Mar-22	88	74	69	69	69
12	Apr-22	86	76	77	72	76
13	May-22	85	69	66	67	66
14	Jun-22	83	69	67	68	67
15	Jul-22	90	78	72	74	72
16	Aug-22	89	71	73	73	75
17	Sep-22	87	72	70	71	71
18	Oct-22	83	58	54	54	55
Average result		**86**	**71**	**69**	**70**	**70**
Standard deviation		**3**	**4**	**3**	**3**	**4**
Run time (minutes)		**00:23**	**04:22**	**00:17**	**00:16**	**00:14**

Table 8. The results (NDCG@8)

Trial no.	Test month	First models (ingredients prediction)	Second models (medicines prediction)	Benchmark model (pointwise) mean squared loss	Benchmark model (pairwise) pairwise hinge loss	Benchmark model (listwise) list MLE loss
1	May-21	92	77	75	77	77
2	Jun-21	87	74	72	72	72
3	Jul-21	87	77	76	77	76
4	Aug-21	90	73	71	71	71
5	Sep-21	84	66	65	67	68
6	Oct-21	88	71	72	73	72
7	Nov-21	87	76	73	75	75
8	Dec-21	90	80	77	77	79
9	Jan-22	81	66	68	67	67
10	Feb-22	91	77	70	69	69
11	Mar-22	89	75	72	71	71
12	Apr-22	88	78	79	77	78
13	May-22	87	71	68	69	69
14	Jun-22	86	72	69	69	69
15	Jul-22	92	81	75	76	76
16	Aug-22	91	75	74	76	77
17	Sep-22	88	73	73	72	73
18	Oct-22	85	62	57	58	58
Average result		**88**	**74**	**71**	**72**	**72**
Standard deviation		**3**	**4**	**4**	**4**	**4**
Run time (minutes)		**00:29**	**04:47**	**00:21**	**00:20**	**00:18**

Figure 7 illustrates a comparative analysis of the NDCG@8 versus NDCG@3 results across all models. The results show consistency between NDCG@3 and NDCG@8 scores across the 18 trials, with NDCG@8 scores consistently higher than NDCG@3 scores. This is expected, as an increase in k allows the algorithm to consider more items in the ranking, potentially leading to a higher score if the additional items are relevant to the user. However, if the additional items are irrelevant, the score remains unchanged.

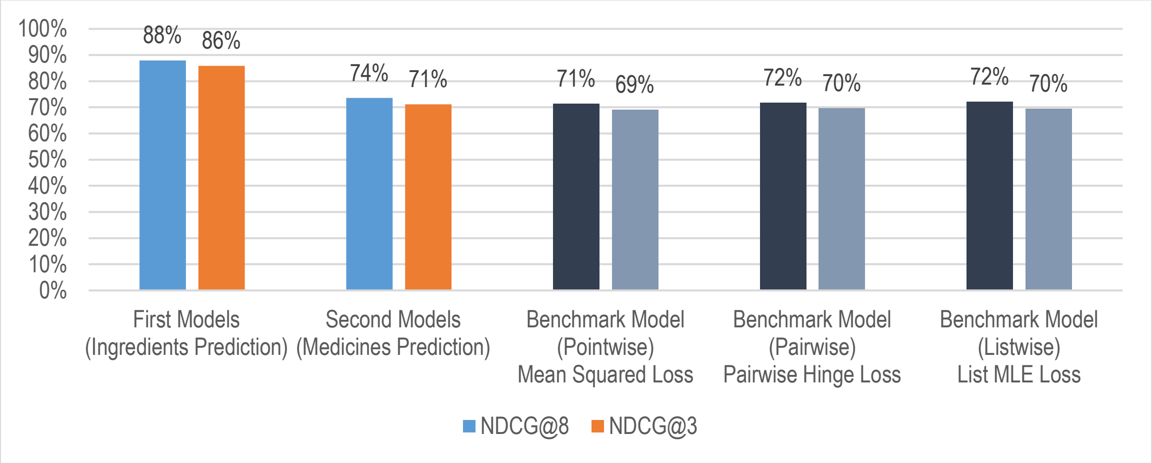


Figure 6. Average NDCG@8 vs average NDCG@3 results

As indicated in Table 8, the average NDCG@8 score across all trials for the ingredient prediction model is 88%, with a standard deviation of 3%. For the medicine prediction model, the average NDCG@8 score is 74%, with a standard deviation of 4%. These results demonstrate that the models are reasonably robust, yielding stable results across different trials. Additionally, our approach outperformed the benchmark models by an average of 3% over the pointwise model and 2% over the pairwise and listwise models. However, it is noteworthy that the benchmark models outperformed our approach in some trials or test months. The benchmark models also exhibited a consistent standard deviation of 4%, similar to our approach.

The improved performance of our models came with the trade-off of longer runtime, averaging around 5 hours per trial, compared to approximately 20 minutes per trial for the benchmark models. To assess the statistical significance of our results, we conducted a paired two-sample T-test with an alpha of 5%. Table 9 presents the significance test results between our approach and the pointwise model, Table 10 between our approach and the pairwise model, and Table 11 between our approach and the listwise model. The test was conducted over 18 trials, with results evaluated using NDCG@8. A two-sample t-test was applied to compare the proposed approach against each of the benchmark methods. The findings revealed a statistically significant improvement in performance, with p-values of 0.7e-3, 3.5e-3, and 1.2e-2 against the pointwise, pairwise, and listwise approaches, respectively. These p-values indicate that the proposed approach outperforms the benchmark methods in ranking medicines based on disease names, with the difference in performance being statistically significant.

Furthermore, the results suggest a reasonable correlation between disease treatments and medicine ingredients. Within each ingredient, certain medicines consistently dominate in preference, making them more favorable choices for both doctors and insurance companies. This dominance likely accounts for the relatively strong performance in medicine prediction when selecting the medicine with the highest predicted probability for each of the top eight predicted ingredients per disease.

Table 9. Significance test between our approach and the pointwise model results

Statistical item	Medicines prediction model (proposed approach)	Benchmark model (pointwise) Mean Squared Loss
Mean	73.56	71.44
Variance	25.08	25.20
Observations	18	18
Pearson correlation	89%	
Hypothesized mean difference	0	
df	17	
t Stat	3.77	
P(T<=t) one-tail	0.7^{-3}	
t Critical one-tail	1.74	
P(T<=t) two-tail	1.5^{-3}	
t Critical two-tail	2.11	

Table 10. Significance test between our approach and the pairwise model results

Statistical item	Medicines prediction model (proposed approach)	Benchmark model (pairwise) pairwise hinge loss
Mean	73.56	71.83
Variance	25.08	24.5
Observations	18	18
Pearson correlation	87%	
Hypothesized mean difference	0	
df	17	
t Stat	2.90	
P(T<=t) one-tail	5^{-3}	
t Critical one-tail	1.74	
P(T<=t) two-tail	9.8^{-3}	
t Critical two-tail	2.11	

Table 11. Significance test between our approach and the listwise model results

Statistical item	Medicines prediction model (proposed approach)	Benchmark model (listwise) list MLE loss
Mean	73.56	72.06
Variance	25.08	26.06
Observations	18	18
Pearson correlation	87%	
Hypothesized mean difference	0	
df	17	
t Stat	2.45	
P(T<=t) one-tail	12^{-3}	
t Critical one-tail	1.74	
P(T<=t) two-tail	25^{-3}	

While analyzing the results, variations were observed across the 18 trials. NDCG is a widely used evaluation metric in information retrieval and recommendation systems, with scores close to 1 indicating that the system has placed highly relevant items at the top of the list, which is the desired outcome. However, these variations in reported NDCG results could be attributed to various factors that warrant further investigation.

- *Different test sets:* The NDCG results may vary if different test sets are used for evaluation, even if the same recommendation algorithm is applied.
- *Different parameter settings:* The NDCG results may also vary if different parameter settings are used for the same algorithm.
- *Different ranking methods:* The NDCG results may also differ based on the ranking method used, such as the use of different loss functions or different optimization algorithms.
- *Different implementations:* The NDCG results can also vary if different implementations of the same algorithm are used.

Since we used similar parameter settings in the 18 trials for the different algorithms and looking into the results in the 18 trials, we can see that the variations are due to the difference in the test datasets for all models since the trend in results is similar for the 18 trials. For example, test months October 2022, September 2021, and January 2022 showed the lowest NDCG results for all of them. This confirms that the variations are due to test data.

FINDINGS AND DISCUSSION

Attempting to predict medicines directly from diseases yielded poor results due to the lack of a strong correlation between disease-related features and medicine codes. In our approach, we sought to emulate the clinical practice by first identifying the most relevant active ingredients for each disease. These predicted ingredients were then used in conjunction with disease features to forecast the corresponding medicines. This method outperformed direct medicine prediction primarily because it incorporated the powerful new feature of the ingredient.

The model's performance was influenced by the presence of diseases associated with multiple ingredients and medicines, each with similar usage ratios or probabilities. For example, in certain trials, the disease "Pain in throat and chest" (ICD code "R07") was treated with three ingredients – PARACETAMOL, IBUPROFEN, and HERBAL COUGH SYRUP – each having a usage ratio of 26.3%. Similarly, for this disease, two medicines, "JOS-PAN SYRUP (120ML)" and "HELIX COUGH SYRUP (125ML)," shared an identical usage ratio of 5.2%. This scenario may lead to arbitrary yet accurate medicine predictions, potentially lowering metric scores, as the model selects only one medicine per disease-ingredient combination and disregards the others. Figure 8 illustrates the number of diseases with multiple high-frequency medicines in each trial's training dataset.

Another factor impacting the NDCG score is the number of distinct medicines historically used per disease. When more medicines are used per disease, the likelihood of the most frequent medicines varying between trials increases. For instance, a disease may be treated more frequently with one medicine in one month and another in the following month. Analysis of the datasets from the 18 trials revealed that, on average, the most frequent medicine per disease in the test dataset differed from that in the training dataset for approximately 24% of the diseases. Figure 9 shows the percentage of diseases in the test dataset for which the most frequent medicine differs from that in the training dataset.

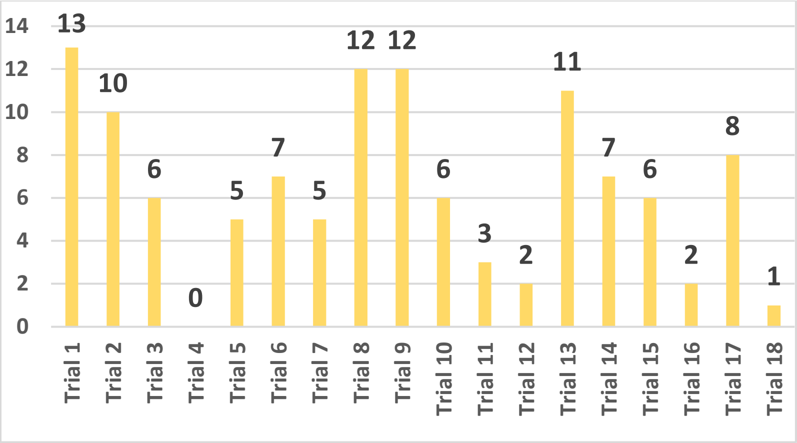


Figure 7. Number of diseases with multiple most frequent medicines per trial

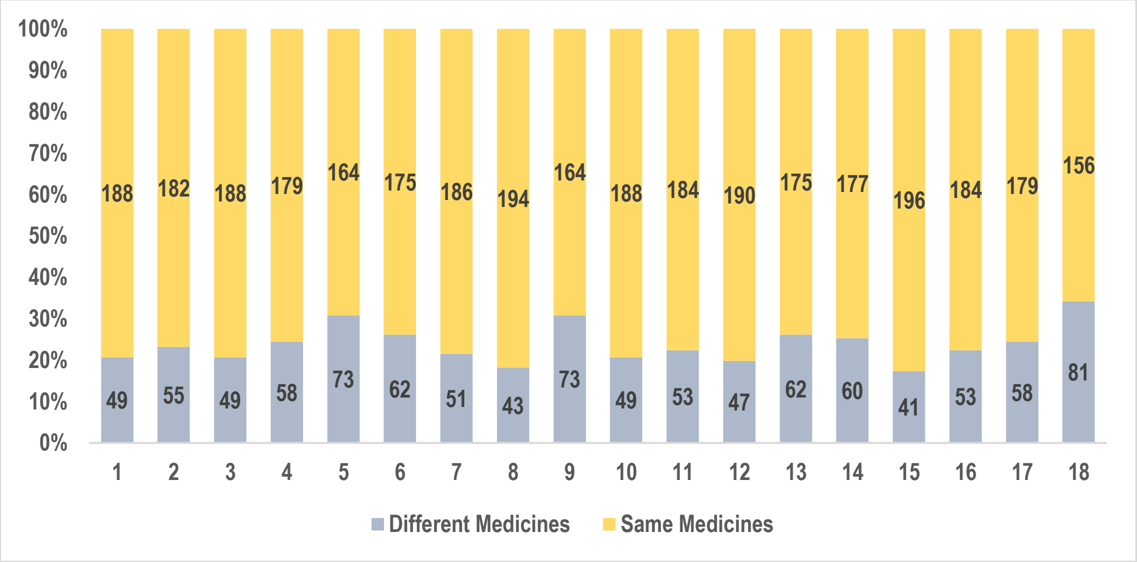


Figure 8. Diseases with different most frequent medicines in test dataset (counts)

The number of available medicines per ingredient also influences the NDCG score. A higher number of medicines for the same ingredient increases the potential for prescription variations, leading to a lower NDCG score. Data showed that the average number of medicines per ingredient is around nine, with a median of five and a maximum of 52 for the active ingredient “CHOLECALCIFEROL (VITAMIN D3).” Figure 10 presents a boxplot showing the distribution of distinct medicine counts per ingredient. These findings indicate that medicine prediction models exhibit less stability than ingredient prediction models, primarily due to the high variability in medicines containing the same ingredient. This variability arises naturally from differences in medicine specifications based on the manufacturing company, price, unit type, dosage form, dosage unit, package type, and package size.

The variability in medicine prediction results across trials can be attributed to several factors. Beyond the aforementioned issue of medicine variations, it was observed that the probabilities of ingredient and medicine usage for the same diseases fluctuate from month to month. This is especially true for rare or low-frequency diseases with few associated ingredients, where one medicine may have a higher usage probability in one month and a lower probability in another, resulting in considerable deviations in outcomes. Finally, it is worth noting that implementing this model in an insurance company could potentially reduce medicine-related paid claims by approximately 5%, as this percentage represents the historical waste from unnecessary medications identified through thorough analysis. Additionally, automating the medical claims submission process could further reduce expenses.

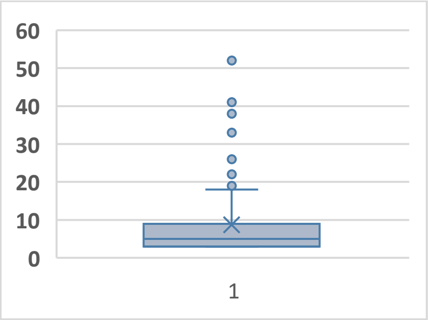


Figure 9. Distribution of the number of distinct medicines per ingredient

THREATS TO VALIDITY

A key threat to the validity of this study is the availability and quality of the data features. The analysis relies solely on ICD-9 disease names and their parent categories within the ICD hierarchy for ranking medicines. This limited information may not be sufficient to accurately rank medicines based on their relevance to specific diseases. Incorporating additional features such as side effects, efficacy, and patient preferences could offer a more holistic view of medicine relevance and potentially enhance the accuracy of the rankings. Moreover, data quality is another critical factor that could influence the study's outcomes. Missing information or data errors could adversely affect the performance of the LTR approaches. Therefore, the findings of this study are constrained by the available data and should be interpreted with caution.

The process of ranking a list of available medicines for a patient involves multiple considerations, each influencing the final decision to varying degrees. Below is a list of some key factors that may impact medicine selection decisions: the diseases and symptoms diagnosed by the doctors.

- Patient profiles include age, gender, physical characteristics, and medication tolerance.
- Patients' case-specific conditions include the occurrence number of the same disease within a specified time frame, possible side effects from certain medications, and existing medical conditions.
- Medicine characteristics include active ingredients and their strength, efficacy and side effects, and drug interaction.

As previously mentioned, a significant threat to the validity of this study lies in the availability and quality of the data features. Our dataset lacks several critical factors, including patient physical characteristics, medication tolerance, existing medical conditions, drug efficacy, side effects, and drug interactions. The absence of these features undoubtedly impacts the ranking performance of both our proposed approach and the benchmark approach. Unfortunately, the missing data cannot be retrieved, so we have accepted this limitation.

Another challenge concerns the quality of the available data. Some historical data has been improperly recorded or accumulated at a higher level of granularity, reducing its applicability in data science approaches. For example, insurance claim entries often aggregate non-chronic medications under broad categories like "local medicine" or "foreign medicine." Moreover, not all data is captured accurately or completely. Crucial information such as age and body mass index (BMI) is frequently missing or incorrectly recorded, with missing values ratios of 64%, 43%, and 89% for age, gender, and BMI, respectively.

Handling missing data is a critical aspect of data preprocessing and can significantly influence machine learning performance. In our study, the gender, age, and BMI features for patients with specific diseases exhibited a high percentage of null values. This is a common issue in real-world datasets and may result from data entry errors or incomplete data collection. Several methods exist for handling missing data, including imputation and deletion techniques. However, given the high proportion of missing values, we opted to exclude the gender, age, and BMI features from our analysis. Imputing these values could introduce additional bias and errors into the dataset. Deleting missing values, a common approach in machine learning simplifies the analysis and is based on the assumption that the missing values are random and not related to the outcome of interest.

In summary, we decided to exclude the gender, age, and BMI features to ensure the quality and reliability of the data used in this study, acknowledging that this decision reduces the dataset's size but enhances the overall robustness of the analysis.

CONCLUSION

Healthcare is a critical domain involving various stakeholders, including insurance companies. Accurate prediction and ranking of potential treatments for each medical case can significantly benefit medical providers, medical coders, and insurers by improving treatment decisions and compensation processes.

Using a confidential dataset from an insurance company, we proposed a pointwise ranking approach. This approach involved developing a process pipeline that splits the treatment ranking for each disease into two prediction tasks: first, ingredient prediction, followed by medicine prediction based on the predicted ingredients. Our method utilizes deep learning models that handle each label separately, with distinct model branches for each label. These models employ regression techniques to predict ingredient and medicine probabilities based on historical data.

To evaluate the robustness of our approach, we conducted 18 trials, treating each monthly cohort of treatments as a separate testing period. We compared our results with a benchmark approach from the Information Retrieval domain, which uses LibSVM data representation. The benchmark approach involved running the same number of trials with three LTR algorithms: pointwise, pairwise, and listwise. Our approach demonstrated superior performance, achieving an average NDCG@8 score of 74%, compared to 71%, 72%, and 72% for the pointwise, pairwise, and listwise benchmark models, respectively.

Our method relies solely on historical data from patients' pharmacy visits, including information on diseases, ingredients, and medicines. Consequently, newly introduced medicines do not appear in the top-ranked results until they gain prominence over time. Given the continuous evolution of the pharmaceutical industry, future work should consider incorporating additional features, such as time, medicine company profiles, market data, and other relevant factors.

REFERENCES

Burges, C. J. (2010). *From ranknet to lambdarank to lambdamart: An overview*. Microsoft Research Technical Report MSR-TR-2010-82.

Burges, C. J. C., Ragno, R., & Le, Q. V. (2006). Learning to rank with nonsmooth cost functions. In B. Schölkopf, J. Platt, & T. Hofmann (Eds.), *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference* (pp. 193-200). MIT Press. https://doi.org/10.7551/mitpress/7503.003.0029

Burges, C. J. C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., & Hullender, G. (2005). Learning to rank using gradient descent. *Proceedings of the 22nd International Conference on Machine Learning* (pp. 89-96). Association for Computing Machinery. https://doi.org/10.1145/1102351.1102363

Cao, Z., Qin, T., Liu, T. Y., Tsai, M. F., & Li, H. (2007). Learning to rank: From pairwise approach to listwise approach. *Proceedings of the 24th International Conference on Machine Learning* (pp. 129-136). Association for Computing Machinery. https://doi.org/10.1145/1273496.1273513

Chakradhar, S. (2017). Predictable response: Finding optimal drugs and doses using artificial intelligence. *Nature Medicine*, *23*, 1244-1248. https://doi.org/10.1038/nm1117-1244

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, *2*(3), Article 27. https://doi.org/10.1145/1961189.1961199

Electronic Health Solutions. (2019). *Hakeem National E-Health Program*. https://ehs.com.jo/hakeem-program

Freund, Y., Iyer, R., Schapire, R. E., & Singer, Y. (2003). An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, *4*, 933-969.

Gerdes, H., Casado, P., Dokal, A., Hijazi, M., Akhtar, N., Osuntola, R., Rajeeve, V., Fitzgibbon, J., Travers, J., Britton, D., Khorsandi, S., & Cutillas, P. R. (2021). Drug ranking using machine learning systematically predicts the efficacy of anti-cancer drugs. *Nature Communications*, *12*, Article 1850. https://doi.org/10.1038/s41467-021-22170-8

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT Press.

Haq, H. U., Ahmad, R., & Hussain, S. U. (2017). *Intelligent EHRS: Predicting procedure codes from diagnosis codes*. arXiv:1712.00481. https://doi.org/10.48550/arXiv.1712.00481

Hosseini, A., Chen, T., Wu, W., Sun, Y., & Sarrafzadeh, M. (2018, October). Heteromed: Heterogeneous information network for medical diagnosis. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (pp. 763-772). https://doi.org/10.1145/3269206.3271805

Jansen, B. J., Spink, A., & Taksa, I. (Eds.). (2008). *Handbook of research on web log analysis*. IGI Global. https://doi.org/10.4018/978-1-59904-974-8

Jin, J., & Garg, H. (2023). Intuitionistic fuzzy three-way ranking-based TOPSIS approach with a novel entropy measure and its application to medical treatment selection. *Advances in Engineering Software*, *180*, 103459. https://doi.org/10.1016/j.advengsoft.2023.103459

Joachims, T. (2002, July). Optimizing search engines using clickthrough data. *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 133-142). Association for Computing Machinery. https://doi.org/10.1145/775066.775067

Kelly, C. J., Karthikesalingam, A., Suleyman, M., Corrado, G., & Dominic King, D. (2019). Key challenges for delivering clinical impact with artificial intelligence. *BMC Medicine*, *17*, Article 195. https://doi.org/10.1186/s12916-019-1426-2

Kumar, S. A., Kumar, A., Dutt, V., & Agrawal, R. (2021, February). Multi model implementation on general medicine prediction with quantum neural networks. Proceedings of the *Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks, Tirunelveli, India,* 1391-1395. https://doi.org/10.1109/ICICV50876.2021.9388575

Lavecchia, A. (2019). Deep learning in drug discovery: Opportunities, challenges and future prospects. *Drug Discovery Today*, *24*(10), 2017-2032. https://doi.org/10.1016/j.drudis.2019.07.006

Levy, J., Vattikonda, N., Haudenschild, C., Christensen, B., & Vaickus, L. (2022). Comparison of machine-learning algorithms for the prediction of current procedural terminology (CPT) codes from pathology reports. *Journal of Pathology Informatics*, *13*, 100165. https://doi.org/10.4103/jpi.jpi_52_21

Li, M., Yao, L., Wang, Q., Wang, X., & Yang, K. (2022). Ranking treatments in the network meta-analysis should consider the certainty of evidence. *The Lancet Gastroenterology & Hepatology*, *7*(4), 287-288. https://doi.org/10.1016/S2468-1253(21)00470-2

Liu, N. N., & Yang, Q. (2008, July). EigenRank: A ranking-oriented approach to collaborative filtering. *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 83-90). Association for Computing Machinery. https://doi.org/10.1145/1390334.1390351

Liu, N. N., Zhao, M., & Yang, Q. (2009). Probabilistic latent preference analysis for collaborative filtering. *Proceedings of the 18th ACM Conference on Information and Knowledge Management* (pp. 759-766). Association for Computing Machinery. https://doi.org/10.1145/1645953.1646050

Lu, X. (2023). Implementation of art therapy assisted by the internet of medical things based on blockchain and fuzzy set theory. *Information Sciences*, *632*, 776-790. https://doi.org/10.1016/j.ins.2023.03.044

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* (pp. 1903-1911). Association for Computing Machinery. https://doi.org/10.1145/3097983.3098088

Meng, Y., Speier, W., Ong, M., & Arnold, C. W. (2020). HCET: Hierarchical Clinical Embedding with Topic modeling on electronic health records for predicting future depression. *IEEE Journal of Biomedical and Health Informatics*, *25*(4), 1265-1272. https://doi.org/10.1109/JBHI.2020.3004072

Miyachi, Y., Ishii, O., & Torigoe, K. (2023). Design, implementation, and evaluation of the computer-aided clinical decision support system based on learning-to-rank: Collaboration between physicians and machine learning in the differential diagnosis process. *BMC Medical Informatics and Decision Making, 23*, Article 26. https://doi.org/10.1186/s12911-023-02123-5

Norori, N., Hu, Q., Aellen, F. M., Faraci, F. D., & Tzovara, A. (2021). Addressing bias in big data and AI for health care: A call for open science. *Patterns*, *2*(10), 100347. https://doi.org/10.1016/j.patter.2021.100347

Pasumarthi, R. K., Bruch, S., Wang, X., Li, C., Bendersky, M., Najork, M., Pfeifer, J., Golbandi, N., Anil, R., & Wolf, S. (2019). TF-Ranking: Scalable tensorflow library for learning-to-rank. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 2970-2978). Association for Computing Machinery. https://doi.org/10.1145/3292500.3330677

Rehman, A., Naz, S., & Razzak, I. (2022). Leveraging big data analytics in healthcare enhancement: Trends, challenges and opportunities. *Multimedia Systems*, *28*(4), 1339-1371. https://doi.org/10.1007/s00530-020-00736-8

Reig, M., Forner, A., Rimola, J., Ferrer-Fàbrega, J., Burrel, M., Garcia-Criado, Á., Kelley, R. K., Galle, P. R., Mazzaferro, V., Salem, R., Sangro, B., Singal, A. G., Vogel, A., Fuster, J., Carmen Ayuso, C., & Bruix, J. (2022). BCLC strategy for prognosis prediction and treatment recommendation: The 2022 update. *Journal of Hepatology*, *76*(3), 681-693. https://doi.org/10.1016/j.jhep.2021.11.018

Rendle, S., Freudenthaler, C., Gantner, Z., & Schmidt-Thieme, L. (2012). *BPR: Bayesian personalized ranking from implicit feedback*. arXiv:1205.2618. https://arxiv.org/abs/1205.2618

Ru, X., Ye, X., Sakurai, T., & Zou, Q. (2022). NerLTR-DTA: Drug-target binding affinity prediction based on neighbor relationship and learning to rank. *Bioinformatics*, *38*(7), 1964-1971. https://doi.org/10.1093/bioinformatics/btac048

Scichilone, R., & Giannangelo, K. (2013). *International Classification of Diseases (ICD) and standard clinical reference terminologies: A 21st century informatics solution*. http://www.who.int/classifications/en/

Shi, Y., Karatzoglou, A., Baltrunas, L., Larson, M., Hanjalic, A., & Oliver, N. (2012a). TFMAP: Optimizing MAP for top-n context-aware recommendation. *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 155-164). Association for Computing Machinery. https://doi.org/10.1145/2348283.2348308

Shi, Y., Karatzoglou, A., Baltrunas, L., Larson, M., Oliver, N., & Hanjalic, A. (2012b). CLiMF: Learning to maximize reciprocal rank with collaborative less-is-more filtering. *Proceedings of the Sixth ACM Conference on Recommender Systems* (pp. 139-146). Association for Computing Machinery. https://doi.org/10.1145/2365952.2365981

Shi, Y., Larson, M., & Hanjalic, A. (2010). List-wise learning to rank with matrix factorization for collaborative filtering. *Proceedings of the Fourth ACM Conference on Recommender Systems* (pp. 269-272). Association for Computing Machinery. https://doi.org/10.1145/1864708.1864764

Subotin, M., & Davis, A. (2014). A system for predicting ICD-10-PCS codes from electronic health records. In K. Cohen, D. Demner-Fushman, & J. Tsujii (Eds.). *Proceedings of BioNLP* (pp. 59-67). Association for Computational Linguistics. https://doi.org/10.3115/v1/W14-3409

Subrahmanya, S. V. G., Shetty, D. K., Patil, V., Hameed, B. Z., Paul, R., Smriti, K., Naik, N., & Somani, B. K. (2022). The role of data science in healthcare advancements: Applications, benefits, and future prospects. *Irish Journal of Medical Science*, *191*, 1473-1483. https://doi.org/10.1007/s11845-021-02730-z

Torfi, A., Fox, E. A., & Reddy, C. K. (2022). Differentially private synthetic medical data generation using convolutional GANs. *Information Sciences*, *586*, 485-500. https://doi.org/10.1016/j.ins.2021.12.018

Vaishya, R., & Misra, A. (2022). International rankings of Diabetes and Metabolic diseases related journals in comparison to other medical journals from India. *Diabetes & Metabolic Syndrome: Clinical Research & Reviews*, *16*(7), 102559. https://doi.org/10.1016/j.dsx.2022.102559

Weimer, M., Karatzoglou, A., Le, Q., & Smola, A. (2007). CoFi$^{\text{RANK}}$: Maximum margin matrix factorization for collaborative ranking. *Advances in Neural Information Processing Systems*, *20*, 1593-1600.

Yang, S., Zheng, X., Xiao, Y., Yin, X., Pang, J., Mao, H., Wei, W., Zhang, W., Yang, Y., Haifeng Xu, H., Li, M., & Zhao, D. (2021). Improving Chinese electronic medical record retrieval by field weight assignment, negation detection, and re-ranking. *Journal of Biomedical Informatics*, *119*, 103836. https://doi.org/10.1016/j.jbi.2021.103836

Yang, Y., Siau, K., Xie, W., & Sun, Y. (2022). Smart health: Intelligent healthcare systems in the metaverse, artificial intelligence, and data science era. *Journal of Organizational and End User Computing (JOEUC)*, *34*(1), 1-14. https://doi.org/10.4018/JOEUC.308814

Zeng, D., Peng, R., Jiang, C., Li, Y., & Dai, J. (2022). CSDM: A context-sensitive deep matching model for medical dialogue information extraction. *Information Sciences*, *607*, 727-738. https://doi.org/10.1016/j.ins.2022.05.099

Zhang, H., Zhang, K., Chen, Y., & Ma, L. (2022). Multi-objective two-level medical facility location problem and tabu search algorithm. *Information Sciences*, *608*, 734-756. https://doi.org/10.1016/j.ins.2022.06.083

Zhou, H., Cao, H., Matyunina, L., Shelby, M., Cassels, L., McDonald, J. F., & Skolnick, J. (2020). MEDICASCY: A machine learning approach for predicting small-molecule drug side effects, indications, efficacy, and modes of action. *Molecular Pharmaceutics*, *17*(5), 1558-1574. https://doi.org/10.1021/acs.molpharmaceut.9b01248

Zhu, J., Wang, J., Wang, X., Gao, M., Guo, B., Gao, M., Liu, J., Yu, Y., Wang, L., Kong, W., An, Y., Liu, Z., Sun, X., Huang, Z., Zhou, H., Zhang, N., Zheng, R., & Xie, Z. (2021). Prediction of drug efficacy from transcriptional profiles with deep learning. *Nature Biotechnology*, *39*, 1444-1452. https://doi.org/10.1038/s41587-021-00946-z

AUTHORS

Maher Farouqa is a postgraduate student at Princess Sumaya University for Technology. His research interests include data science, machine learning, and big data analytics. He is currently working in an insurance company as a data scientist.

Mohammad Azzeh is a professor of Data Science at Princess Sumaya University for Technology. Dr Azzeh holds a PhD in computing from the University of Bradford, UK, and an MSc in Software Engineering from the University of the West of England, UK. He is currently working as a faculty staff member in the Data Science department. His research interests focus on data science, machine learning, data mining, empirical software engineering, and mining software repositories. Dr Azzeh is an invited referee for high-quality journals and a PC member of international conferences. He was a guest editor in the Journal of Neural Computing and Applications (Springer). Dr. Azzeh was conference chair of the 7th and 8th International Conferences of Computer Science and Information Technology (2016 and 2018). Dr Azzeh co-organized and co-chaired several special sessions/workshops: Computational Intelligence Applications in Software Engineering (CIASE 2013) at the 3rd IEEE International Conference on Communications and Information Technology (ICCIT 2013), Machine Learning for Predictive Models (MLPM 2013, 2014) at the IEEE International Conference on Machine Learning and Applications (ICMLA 2013 to 2021). He was also the publicity chair of the CSIT 2014 conference. Dr. Azzeh published over 40 research articles in reputable journals and conferences such as IET Software, Software: Evolution & Process, Empirical Software Engineering Applied Soft Computing and Systems & Software, Science of Computer Programming, and Journal of Software Quality.