Computers in Biology and Medicine 178 (2024) 108721

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/compbiomed

Building a challenging medical dataset for comparative evaluation of classifier capabilities

Berat Bozkurt, Kerem Coskun, Gokhan Bakal *

Department of Computer Engineering, Abdullah Gul University, Barbaros, Erkilet Blvd. Sumer Campus, Kayseri, 38080, Turkey

ARTICLE INFO

Dataset link: https://shorturl.at/lotEV

Keywords:
Text mining
Classification
Machine learning
Deep learning

ABSTRACT

Since the 2000s, digitalization has been a crucial transformation in our lives. Nevertheless, digitalization brings a bulk of unstructured textual data to be processed, including articles, clinical records, web pages, and shared social media posts. As a critical analysis, the classification task classifies the given textual entities into correct categories. Categorizing documents from different domains is straightforward since the instances are unlikely to contain similar contexts. However, document classification in a single domain is more complicated due to sharing the same context. Thus, we aim to classify medical articles about four common cancer types (*Leukemia*, *Non-Hodgkin Lymphoma*, *Bladder Cancer*, and *Thyroid Cancer*) by constructing machine learning and deep learning models. We used 383,914 medical articles about four common cancer types collected by the PubMed API. To build classification models, we split the dataset into 70% as training, 20% as testing, and 10% as validation. We built widely used machine-learning (Logistic Regression, XGBoost, CatBoost, and Random Forest Classifiers) and modern deep-learning (convolutional neural networks - CNN, long short-term memory - LSTM, and gated recurrent unit - GRU) models. We computed the average classification performances (precision, recall, F-score) to evaluate the models over ten distinct dataset splits. The best-performing deep learning model(s) yielded a superior **F1 score of 98%**. However, traditional machine learning models also achieved reasonably high **F1 scores, 95%** for the worst-performing case. Ultimately, we constructed multiple models to classify articles, which compose a `hard-to-classify` dataset in the medical domain.

1. Introduction

Before the digitalization era began, information about all processes, including patient records, bank accounts, and the news, was processed by manual human effort. Starting with the early phases of digitalization, data-related operations, such as hospital appointments and communication habits, have been transformed into more convenient and faster forms. For instance, nowadays, people rarely prefer sending letters to others for communication instead of using real-time messaging applications. Nevertheless, the digitalization convenience caused a critical hassle, processing massive data bulks produced by many sources, from wearable devices to social media platforms, to discover new knowledge [1]. Since processing that large volume of data is impractical by manual methods, computational methodologies, such as data mining, machine learning (ML), and, lately, deep learning (DL), are densely employed in various fields like marketing, finance, biomedical sciences, and entertainment [2–5].

Considering all the unique problems regarding data processing, one of the widely studied problems is classification. Technically, classification is labeling test instances with categorical labels. Here, the test examples can be images (e.g., tumor or human face pictures) and textual elements (e.g., articles, social media posts, or e-mails). Concerning text/document classification, the classification models can be constructed for many purposes, including sentiment analysis [6], fraud detection [7], and disease/drug discovery [8,9]. Traditional methods of categorizing medical articles, such as keyword-based searching and manual review processes, are susceptible to inherent limitations that compromise their accuracy and efficiency. Keyword-based searching often relies on predefined terms or phrases, which may fail to capture the complexity and nuance of medical literature. Moreover, manual review processes are time-consuming and labor-intensive, subject to human error and inconsistency in interpretation. These limitations hinder the comprehensive and reliable categorization of medical articles, particularly in domains with intricate terminology and diverse contextual nuances [10]. As a result, there is a pressing need for more sophisticated approaches, such as natural language processing (NLP), to overcome these challenges and facilitate more accurate and efficient information retrieval and categorization in the medical research domain.

* Corresponding author.
E-mail address: gokhan.bakal@agu.edu.tr (G. Bakal).

https://doi.org/10.1016/j.compbiomed.2024.108721
Received 25 January 2024; Received in revised form 28 May 2024; Accepted 5 June 2024
Available online 19 June 2024

In this study, we intend to classify medical articles consisting of, particularly, four common cancer diseases. Our principal point in selecting a medical dataset is to have a relatively complicated classification experiment. Beyond that, we deliberately narrowed the articles' subject span by picking only common cancer types because we wanted to solve an even more challenging problem. By doing that, we curated a `hard-to-classify` medical dataset for our classification experiments. The novelty of our study primarily lies in **the creation of a compact and challenging-to-classify dataset**, which is intended for benchmarking applications.

In the classification experiments, we constructed four ML models and three DL models utilizing the hard-to-classify cancer dataset. Following that, we computed the performance scores and discussed them to evaluate the models. The key contributions of this study are mentioned below:

- We created a compact and `hard-to-classify` medical dataset consisting of abstract sections in the medical articles about four common cancer diseases to have an even more challenging classification problem.
- We constructed multiple ML models which are using well-known algorithms, including Random Forest, Logistic Regression, XGBoost, and CatBoost; additionally, we built powerful and popular DL models containing CNN, LSTM, and GRU architectures.
- We subjected our `hard-to-classify` dataset to widely utilized ML and DL models to compare model performances on a relatively troublesome classification problem.

The rest of the paper is formed as follows. Section 2 briefly mentions relevant literature and background information, while Section 3 explains the dataset used in the experiments. Section 4 introduces the methodologies utilized during the model constructions with the experimental details. Section 5 demonstrates the obtained results and evaluates the model performances. Ultimately, Section 6 concludes the presented effort and gives a comprehensive summary.

2. Background & related studies

Due to the exponentially growing digital data, analyzing/classifying huge document volumes with manual efforts is infeasible. Hence, computational systems utilizing machine learning-based techniques are employed to automate manual examinations on large datasets in various domains. For instance, Bakal and Kavuluru [9] used ML models to discover unknown treatments and causative relations over a massive biomedical knowledge graph. From the social media analysis perspective, Jiang et al. [11] conducted experiments employing the MapReduce parallel processing model to solve the friend recommendation problem on large social networks.

Text classification is the task of assigning appropriate label(s) to the test document examples. As a milestone, Borko and Bernick [12] showed that a factor analysis-based approach can be utilized for automatic text classification. Then, Liang [13] showed that the Support Vector Machine (SVM) model (with a one-vs-all infrastructure) is a convenient way of classifying web pages. [14] also demonstrated that a modified version of SVM, SVM one-class, outperformed neural network approaches. Researchers also examined the power of text/document classification over social media posts to predict users' psychological situations [15,16]. Jiang et al. [17] proved that deep learning architectures perform even better in technical document classification. Plus, Behera et al. [18] analyzed the classification performances of modern deep learning-based architectures and usual ML approaches on various datasets. Eventually, they showed that ML algorithms underperformed compared to deep learning-based models. Blanco et al. [19] worked on the multi-label clinical document classification problem using neural network models. Similarly, Sovrano et al. [20] demonstrated that deep learning architectures boosted by Term Frequency - Inverse Document Frequency (TF-IDF) similarities achieve better classification scores.

The primary objective of this study is to construct a compact hard-to-classify dataset and to examine the classification strength of powerful machine learning models (Logistic regression, XGBoost, CatBoost, and Random Forest classifiers) and popular deep learning models, including CNN, LSTM, and GRU architectures. We intentionally picked these classifier algorithms because the constructed ML models utilize n-gram features and are powerful at classification. Similarly, the DL models we build are successful in capturing contextual associations between words, yielding better classification performances.

Table 1
Document distribution in the dataset.

Cancer type	Number of documents
Leukemia	226,552
Non-Hodgkin Lymphoma	61,440
Bladder Cancer	46,176
Thyroid Cancer	49,746

3. Dataset details

To curate the dataset, we utilized the PyMed python module, which provides convenient access to the PubMed library [21] interface, to pull the medical articles by a given disease name (including Leukemia, Non-Hodgkin Lymphoma, Bladder Cancer, and Thyroid Cancer). As a response to each fetch request, we received a massive list of JSON objects containing data fields regarding the article, such as title, authors, abstract, and keywords. Following this process, the articles are prefiltered as those with abstract sections and available under a single disease type. After the filtration, the statistical distribution of the articles about target diseases is shown in Table 1.

The specific cancer types Leukemia and non-Hodgkin lymphoma were intentionally picked to have an even more challenging classification task compared to the classification of domain-independent documents. This situation happens because both cancer types are blood cancer sub-types, and it is more challenging to distinguish them apart. The word cloud representation extracted over the whole data corpus is illustrated in Fig. 1.

Each word cluster represents each cancer type; thyroid cancer, leukemia, non-Hodgkin lymphoma, and bladder cancer, respectively. When we inspect the word clusters, it is apparent that leukemia and non-Hodking lymphoma data collections share numerous domain-related terms. Regarding the composition rationale of the dataset, we employed a strategic selection process to achieve a balance between dataset compactness and power in this study. We included two relatively similar cancer subtypes (Leukemia and Non-Hodgkin Lymphoma) to capture nuanced differences within a specific cancer class. Additionally, we incorporated two highly dissimilar cancer subtypes (Bladder and Thyroid cancers) to broaden the dataset's scope and enhance its generalizability.

4. Method

Here, the core methodologies are based on various ML and DL models to solve the `hard-to-classify` medical dataset. In Section 4.2, the ML concept is explained, and then the four distinct classification algorithms used are introduced in the following subsections. Then, the DL notion and popular DL models constructed are presented in Section 4.3 and the corresponding subsections.

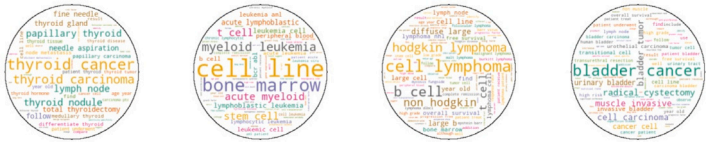

Fig. 1. Word cloud representation extracted from the dataset.

4.1. Data preprocessing & data input format

In our preprocessing pipeline, several essential steps are employed to enhance the quality and efficiency of text data processing. Firstly, we incorporate lemmatization, a technique that reduces words to their base or dictionary form, thereby standardizing vocabulary and improving text normalization. This step aids in reducing lexical variation and ensuring consistency across the dataset, which is particularly valuable in medical text analysis where terminology can vary. Additionally, we implement stopwords removal, which involves filtering out common words such as "the", "is", and "and" that do not contribute significantly to the semantic meaning of the text. By eliminating these irrelevant tokens, we reduce noise and dimensionality in the data, facilitating more focused and accurate analysis. These preprocessing steps collectively contribute to refining the textual data, laying a solid foundation for subsequent analysis and modeling tasks.

For traditional machine learning models, we utilize the TF-IDF vectorizer to transform raw text data into a numerical representation suitable for modeling. The TF-IDF vectorizer computes a numerical value for each word in the document based on its frequency within the document and across the entire corpus, while also down-weighting words that occur frequently across documents. This process results in a sparse matrix where each row corresponds to a document and each column represents a unique word in the corpus, with values indicating the importance of each word in the respective document. By encoding text data in this manner, we convert unstructured text into a structured format that traditional ML algorithms can effectively process for our classification experiments.

In parallel to our preprocessing pipeline for traditional machine learning models, we adopt a tailored approach for deep learning models, specifically LSTM, RNN, and GRU. Our input data undergoes a sequence of transformations to facilitate their compatibility with these architectures. Initially, utilizing the Keras Tokenizer module, we tokenize the text data and limit the vocabulary to the top 50,000 words by ensuring computational efficiency without compromising information richness. Subsequently, sequences of tokens are generated for both training and testing datasets. To standardize input length, we pad these sequences to a maximum length of 220 words for uniformity across samples. Finally, an embedding layer with a dimensionality of 100 is employed to capture semantic relationships between words. These tailored preprocessing steps enable our deep learning models to effectively ingest and interpret textual data to empower them to extract intricate patterns and insights embedded within medical texts. To highlight the most discriminative words in each document collection, we identified and ranked the top 20 words by their TF-IDF scores. These results are presented in Table 2.

Table 2
Top 20 descriptive words ranked by their TF-IDF scores for each class in the document collection.

Bladder Cancer	Thyroid Cancer	Non-Hodgkin Lymphoma	Leukemia
patient	thyroid	cell	cell
bladder	patient	patient	patient
cancer	cancer	lymphoma	leukemia
cell	carcinoma	case	aml
tumor	cell	disease	treatment
carcinoma	nodule	treatment	acute
case	tumor	year	gene
treatment	case	nhl	case
study	ptc	tumor	study
group	study	hodgkin	expression
urinary	year	study	disease
use	disease	survival	use
year	group	non	protein
recurrence	treatment	therapy	therapy
survival	papillary	chemotherapy	year
expression	risk	clinical	myeloid
risk	metastasis	dlbcl	show
tumour	use	diagnosis	marrow
disease	diagnosis	expression	activity
cystectomy	follicular	ebv	result

4.2. Machine learning models

Machine learning is a teaching method for computers to learn from training data without being explicitly programmed by a human. It is a sub-field of the artificial intelligence discipline that involves the development of algorithms and models that enable computers to learn from and make predictions or decisions without human intervention. There are three major branches of machine learning: *supervised learning*, *unsupervised learning*, and *semi-supervised learning*. Supervised learning is the most common type, where a model is trained on labeled (annotated) data, which means that the desired output is already known before the model building [22]. After the learning process is completed, the model can make predictions on new, unseen data elements. Common examples of the supervised learning approach include linear regression, support vector machines, and artificial neural network algorithms.

Unlike the supervised learning approach, unsupervised learning involves training a model on unlabeled data instances where the desired output is unknown. The model is then able to identify patterns or associations in the data [23]. Common examples of the unsupervised learning concept include k-means clustering, principal component analysis, and matrix factorization-based algorithms.

Semi-supervised learning is a machine learning technique that combines labeled and unlabeled data for training a model. Unlike supervised learning, which uses only labeled data, and unsupervised learning, which uses only unlabeled data, semi-supervised learning leverages the strengths of both approaches. The idea behind semi-supervised learning is to leverage large amounts of unlabeled data to improve the model' s performance by incorporating additional information into the learning process. Semi-supervised learning algorithms use distinct techniques, such as self-training, co-training, and generative adversarial networks (GANs) [24], to utilize labeled and unlabeled data together. The primary goal of semi-supervised learning is to learn a model that generalizes well to new, unseen data while improving accuracy and reducing the need for large amounts of labeled data.

In the ML experiments, the raw text elements in the abstract sections are converted into TF−IDF vectors to prepare the textual data input for the models. This process operates by calculating the frequency of each word and weighting them based on their rarity across the entire data corpus. Hence, this procedure allows us to represent the abstracts as numerical vectors that could be used as input for the machine learning models.

4.2.1. Logistic regression algorithm

Logistic Regression (LR) is a statistical method employed for classification tasks, including multi-class classification problems. Technically, it uses a logistic function to model a binary dependent variable, which

is used to predict the probability of an observation belonging to a particular class [25]. The logistic function, also known as the sigmoid function, maps any real-valued number to a value between the range of [0, 1], which can be interpreted as a probability score. The logistic regression classifier is trained to identify the best coefficients for the input features, which are used to make predictions on new/unseen data. Essentially, the classifier assigns an instance to the class with the highest predicted probability during the learning process. Beyond binary classification experiments, it can also be extended to handle multi-class classification problems by training multiple binary classifiers (modeled with the one-vs-rest approach) and combining their results. In the experiments, we used the scikit-learn ML library, which contains the LR algorithm as a python function wrapper.

4.2.2. Random forest algorithm

Random Forest (RF) is a widely-utilized ensemble learning method for classification and regression problems in the machine learning field. It simply combines multiple decision trees to construct a more robust model. In a random forest space, each decision tree is built on a random sample of the data and features, and the final prediction is made by averaging the predictions of all trees. By performing this operation, the learning process reduces the overfitting tendency of individual trees and leads to improved generalization performance. It also includes feature selection as part of the training process, as it splits the data based on the feature that provides the best split according to a pre-defined criterion. This ability allows the model to select a compact and interpretable set of features, which improves the accuracy and interpretability of the model. Additionally, as in gradient-boosting techniques, the random forest algorithm can handle missing values and non-linearly separable data, making it a versatile and widely-employed classifier in various fields. In this experimental study, we used the RF algorithm available from the popular Python-based ML library, scikit-learn, for both binary and multiclass classification problems [26].

4.2.3. XGBoost algorithm

XGBoost (eXtreme Gradient Boosting) is an efficient open-source machine learning algorithm used for classification and regression problems [27]. Technically, it is based on the gradient boosting infrastructure and uses decision trees as its base learning mechanism. XGBoost handles missing values and categorical variables and can perform parallel computation on a system with multiple cores or in a distributed computing environment. The algorithm' s training process involves adding trees one at a time, where each new tree works to correct the mistakes made by the preceding tree. It also includes various optimizations such as regularization, sparsity awareness, and approximate split finding, which can help improve its performance over traditional gradient-boosting algorithms. Overall, we deliberately built an XGBoost classifier to challenge the state-of-the-art deep learning models because it provides considerably high classification performance, and the outcomes are more interpretable against DL models.

4.2.4. CatBoost algorithm

CatBoost is an open-sourced machine learning library developed by Yandex ML team that is used for gradient boosting [28]. It is designed to work with categorical features and handle missing values. It also has built-in support for handling categorical features, which is useful for working with datasets that have a lot of categorical variables. Additionally, it has various features for handling missing values, such as the ability to fill in missing values with the mean or mode of the column. It also has a built-in feature importance calculation that allows users to understand which features are most important in their models for further feature engineering studies. Overall, CatBoost is a powerful tool for building gradient-boosting models on decision trees, particularly when working with datasets that have a lot of categorical features and missing values.

4.3. Deep learning models

Deep learning architectures are neural network models that are composed of multiple layers, with the number of layers varying depending on the specific architecture. Some popular architectures include simple feedforward neural networks, CNNs, and RNNs. Feedforward neural networks are the basic type of deep learning architecture, where the data flows in one direction from input to output. CNNs are commonly used for image/video recognition and textual interpretation tasks, while RNN models are used for studies involving sequential/temporal data. The success of deep learning architectures is largely due to their ability to learn hierarchical representations and contextual associations of data, which allows them to achieve state-of-the-art results on various tasks. In this work, we utilized Keras deep learning library [29] for the deep learning models.

4.3.1. CNN architecture

A CNN is a typical deep learning model first designed for image classification and computer vision tasks. It consists of multiple layers of convolution, activation, pooling, and fully connected layers [30]. The convolution layers apply filters to the input, which are then passed through activation functions such as *tanh* to introduce non-linearity. The pooling layers reduce the dimensionality of the output from the convolution layer. Finally, the fully connected layers perform the final classification based on the features learned by the network. CNNs are highly effective in classification tasks and yield state-of-the-art performance results. This is because it can capture local patterns and relationships in the data, which makes it effective at tasks such as image and text classification.

4.3.2. LSTM architecture

LSTM is a specific type of RNN designed to handle the vanishing gradients problem in traditional RNNs. LSTM networks are capable of learning long-term dependencies by using particular gates to control the flow of information passed through the memory units. These gates called forget gate, input gate, and output gate, determine which information will be discarded or passed through to the next time step. The forget gate decides which data to discard/forget from the memory cell, while the input gate determines which information to add to the memory cell [31]. The output gate produces the final output based on the memory cell state. The representation of the cell structure is demonstrated in Fig. 2. By employing these gates, LSTMs can avoid the vanishing gradient problem and capture long-term dependencies in sequential data. The network trains on the input–output pairs and adjusts the weights of the gates and memory cells to minimize the prediction error. LSTMs have been successful in various sequence-to-sequence tasks such as machine translation, text generation, classification, and stock price prediction.

4.3.3. GRU architecture

GRU is a subtype of recurrent neural network that aims to solve the vanishing gradient problem in traditional RNNs. GRU networks use a gating mechanism to control the flow of information as in the LSTM architecture and decide what information to preserve and to discard from the hidden state. To control the information flow, there are two gates: the reset gate and the update gate. The hidden state is updated at each time step based on the current input and the previous hidden state. The reset and update gates control the flow of information in and out of the hidden state, respectively [32]. The reset gate uses a sigmoid activation function to determine the extent to which the previous hidden state should be forgotten. The update gate uses a sigmoid activation function to determine the extent to which the new information should be added to the hidden state. The new hidden state is a weighted sum of the previous hidden state and the new information, where the weights are determined by the reset and update gates. The hidden state is then passed as input to the next time step. The final

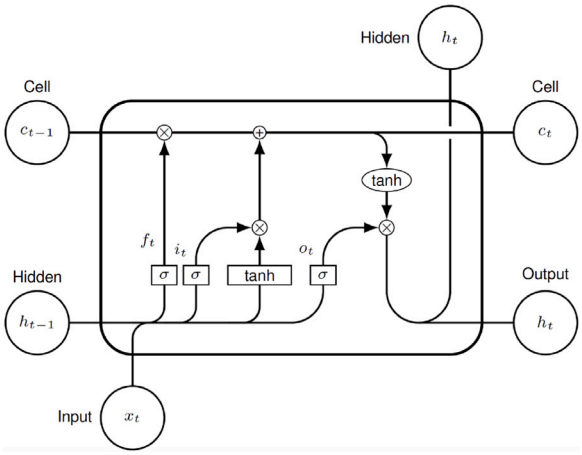

Fig. 2. An example of LSTM cell structure.

output is produced by applying a linear activation function to the final hidden state. GRU models are simpler and computationally more efficient than LSTM systems and have been successful in NLP tasks, such as text classification and language modeling.

4.4. Model configurations & evaluation metrics

Here, we describe the details of the distinct model configurations for ML and DL models under the corresponding subsections. Before the model construction, we split the dataset into training (70%), testing (20%), and validation (10%) subsets. We used the validation set to identify the optimal parameters of the models. Both GridSearch and manual attempt strategies were utilized to perform the hyperparameters tuning.

4.4.1. ML model details

To prepare the input for the ML models, we employed TF-IDF weights represented by `unigram` features. Unigrams are the essential and informative units of text, making them a great choice for representing the input entries to our models. In the LR model, maximum iterations (*max_iter*) and regularization (C) penalty parameters are tuned over the validation set. The *max_iter* parameter is the maximum number of iterations for the solver to converge. The best number of iterations is picked from the list of $[100, 1000]$ values. The C value maintains the inverse of the regularization strength to prevent the overfitting issue. The search space of the C value is as in the list of $[0.001, 0.01, 0.1, 1]$. In the RF model configurations, the number of estimators (*n_estimators*) and *criterion* parameters are optimized by using the validation set. The optimum estimator number is picked from the list of $[25, 50, 75, 100, 125, 150, 750, 1000]$, while the criterion parameter is selected as either *gini* or *entropy*. In the XGBoost model, the $objective$ function is set to $multi : softprob$ to build the model as multi-classification using soft probabilities. Finally, for the CatBoost classifier model, the $loss\_function$ parameter is specified as $MultiClass$ to construct a multi-class classification experiment.

4.4.2. DL model details

As we represented in Table 3, in the CNN model, we have an embedding layer as input, where the input dimension is 30 K words as vocabulary, the embedding size is 100, and the input length (as the sentence length) is 200. Next, a 1D convolution layer ($Conv1D$ with a filter size of 128) and a $GlobalMaxPool1Dlayer$ are used in the network structure to aggregate the underlying relationships among the words. Then, two fully-connected dense layers (unit sizes: 64 and 32) simulate the neural associations, while the final layer has the softmax function with four output units functioning (indicating the number of classes) as the model output. In the LSTM architecture, the network starts with an input embedding layer, where the input dimension is 30 K, the embedding size is 100, and the input length is 200. Then, the next layer consists of the $SpatialDropout1D$ layer to cope with the overfitting issue. In the next layer, an $LSTM$ layer is employed with 100 units. Following that, two fully-connected dense layers (unit sizes: 64 and 32) and the last dense layer with four output units are used in the model. Ultimately, the GRU model also starts with an input embedding layer, where the input length is 220, and the embedding dimension is 200. Then, a GRU layer with 64 units, a dropout of, and $recurrent\_dropout$ rate of 0.2. In the final layer, a fully-connected dense layer with four neural units is employed as the final output. For all DL models, we utilized $sparse\_categorical\_crossentropy$ loss function since the document classes are mutually exclusive.

4.4.3. Performance evaluation metrics

Precision, recall, and F1 score serve as critical metrics in classification tasks since they provide nuanced insights into model performance beyond simple accuracy [33]. Precision measures the proportion of correctly predicted positive cases out of all predicted positives. Recall, on the other hand, calculates the ratio of correctly predicted positive cases to all actual positives, while the F1 score, the harmonic mean of precision and recall, balances these metrics by offering a comprehensive evaluation of a classifier' s performance. The performance scores will be computed based on the formulas presented in Eqs. (1), (2), and (3).

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3}$$

In classification scenarios, where the consequences of false positives (FP) and false negatives (FN) can vary significantly, precision, recall, and F1 score play pivotal roles in gauging the effectiveness and reliability of predictive models.

5. Results & discussion

Considering obtained results, we report the classification performances[1] of traditional machine learning (Logistic Regression, Random Forest, XGBoost, and CatBoost) and deep learning (CNN, LSTM, and GRU) models. We first assess each learning group individually, then evaluate them together using the performance metrics. According to the results reported in Table 4, the XGBoost model outperformed the other models even though all models achieved at least 95% performance for each metric.

Since the performance scores are reasonably high, we also presented the confusion matrix of the best-performing model in Fig. 3. In the confusion matrix, the *0*th index refers the *thyroid cancer*, and *1*st index indicates *leukemia*, while *2*nd index points *non-Hodgkin lymphoma*, and *3*rd represents *bladder cancer* instances.

From the confusion matrix heatmap, we realize that 807 non-Hodking lymphoma examples were predicted as leukemia, while 342 leukemia instances were classified into the non-Hodking lymphoma class. This outcome is not surprising because both cancer types are specific sub-branches of blood cancer, and the corresponding articles share a remarkable number of identical terms and phrases. Thus, it is quite expected that the two highest misclassified example groups are from these cancer types. Another striking point is that 274 bladder

[1] The weighted averages of precision, recall, and f1-scores are reported as the essential assessment scores.

Table 3
DL hyperparameter details for each model.

	CNN	LSTM	GRU
input layer & sizes	*Embedding*, dimension of 100	*Embedding*, dimension of 100	*Embedding*, dimension of 100
hidden layer(s) & sizes	a *Conv1D* layer with a filter size of 128	*a LSTM layer with 100 units*	*a GRU layer with 64 units*
output layer & sizes		*a softmax function with four output units*	
dropout rate		0.2	
loss function		*sparse_categorical_crossentropy*	

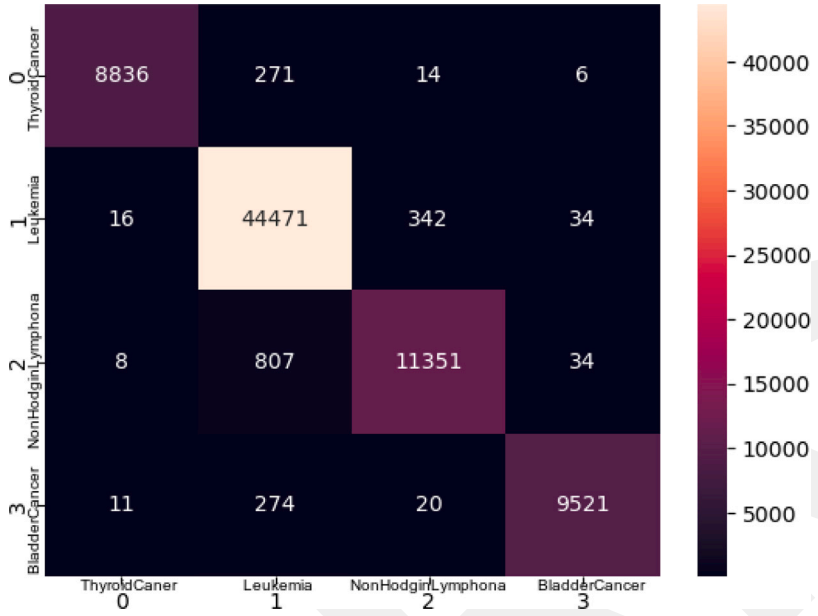

Fig. 3. Confusion matrix with heatmap representation of the XGBoost model.

Table 4
Performance scores of traditional ML models.

Model	Avg. precision	Avg. recall	Avg. F1-score
Logistic Regression	0.97	0.97	0.97
Random Forest	0.95	0.95	0.95
XGBoost	**0.98**	**0.98**	**0.98**
CatBoost	0.96	0.96	0.96

Table 5
Performance scores of deep learning models.

Model	Avg. precision	Avg. recall	Avg. F1-score
CNN	**0.98**	**0.98**	**0.98**
GRU	0.97	0.97	0.97
LSTM	**0.98**	**0.98**	**0.98**

cancer test examples were classified as leukemia. A potential reason for this result can be the association between Chronic lymphocytic leukemia (CLL) and bladder mass [34]. The last impressive outcome is that 271 thyroid cancer instances were predicted with the leukemia label.

As can be noticed from the performance results in Table 5, the CNN and LSTM models outperformed the GRU model even though the difference is only 1% in each performance metric. The potential underlying reasons for this situation are that the LSTM architecture is more powerful and sophisticated by having more parameters to model longer dependencies than the GRU model. Also, LSTM models perform better than GRU on specifically larger datasets, while GRU models are usually more successful on smaller datasets.

Regarding the CNN model, it also yielded nearly superior performance by automatically extracting hidden features from the convolution layers as in the LSTM model. To better illustrate the classification performances based on the number of examples classified into the disease types, the LSTM model' s confusion matrix is shown in Fig. 4. Here, the reason we only showed the confusion matrices for the XGBoost and LSTM models is that both models gave superior performance scores among their model groups.

As reported in the XGBoost model' s confusion matrix, the most challenging classification operation happened between leukemia and non-Hodgkin lymphoma instances. In particular, 531 leukemia test instances were predicted as the non-Hodgkin lymphoma class, while 517 non-Hodgkin lymphoma examples were classified as leukemia class. As expected, this outcome is not surprising again because both diseases are contextually more related to each other compared to the other diseases. The second notable classification difficulty was differentiating examples of leukemia and bladder cancers from each other. We believe this condition was due to the medical relationship between a rare cancer type of chronic lymphocytic leukemia and increased bladder mass [34].

Considering the performance comparison of ML and DL experiments, the DL models built by LSTM and CNN architectures as well as the XGBoost ML model, achieved the highest performance (98%), while the lowest ML model also obtained reasonably high classification achievements (95%). Ultimately, all the experimental models gained high-performance scores even if the target dataset is typically a `hard-to-classify` medical dataset.

In evaluating the generalization capabilities of the classifiers utilized in our study, we found compelling evidence of their robustness and adaptability across diverse datasets. Our analysis revealed that the machine learning and deep learning models, including logistic regression, XGBoost, CNN, LSTM, and GRU, consistently demonstrated strong performance metrics across multiple dataset splits. Furthermore, our findings suggest that these classifiers exhibit promising generalization potential, as evidenced by their ability to maintain high accuracy levels

Fig. 4. Confusion matrix with heatmap representation of the LSTM model.

when applied to unseen data. These results underscore the versatility and efficacy of the employed classifiers, highlighting their suitability for real-world applications beyond the confines of our specific dataset. While we believe that reporting the average precision, average recall, and average F1 scores provides a solid overview of model performance, we have also calculated the Area under the ROC Curve (AUC) scores for the best-performing models, LSTM and XGBoost. The AUC score for the LSTM model is 0.97, and the AUC score for the XGBoost model is 0.95. These results confirm that the classification performances are high and consistent with the previously reported metrics.

6. Conclusion

Text classification is one of the widely studied research problems in many datasets. Multi-class classification is relatively more complicated than binary-class classification due to learning all latent features discriminating each class. However, conducting multi-class classification experiments on single-domain data, as in our proposed work, is even more challenging due to the contextual similarities (*owing to many domain-specific frequent words*) existing in the textual data.

In this paper, we built various machine learning and deep learning models to classify a `hard-to-classify` dataset, which consists of medical articles about four common cancer diseases, and reported their performance results with evaluations. The results show that ML (logistic regression, random forest, XGBoost, and CatBoost classifiers) and DL models (GRU, LSTM, and CNN) performed well in classifying cancer-related articles. The best-performing models are the XGBoost classifier (achieved 98% of precision, recall, and f1-score) as a traditional model, along with the LSTM and CNN models (yielding 98% of each performance score) as deep learning models. Even the lowest-performing models, random forest, and catboost classifiers gained 95% and 96% performance scores, respectively.

7. Limitations and future works

One notable limitation of the current study is the exclusion of state-of-the-art model applications, such as transformer-based models. While it is important to note that these models were not within the primary scope of this work, their absence may limit the study's comprehensiveness. Potential promising avenues for future research of the current work can be explored as:

- **Increasing the number of document classes:** Expanding the number of document classes can provide valuable insights into how the models' performances are affected. Monitoring the impact of class type expansion on performance is an intriguing prospect.
- **Ensemble model construction:** Despite achieving superior performance scores, there is potential for enhancing model robustness. Consideration can be given to building an ensemble model that combines the strengths of the individual models used in the proposed work. Even with the recent F1-scores ranging from 95% to 98%, an ensemble approach may further improve classification accuracy and reliability.
- **Exploiting state-of-the-art models:** As a consideration of future work, we can explore incorporating state-of-the-art transformer-based models, such as GPT (Generative Pre-trained Transformer) and BERT (Bidirectional Encoder Representations from Transformers), into our analysis. These models have shown remarkable capabilities in various natural language processing tasks, and their integration could provide valuable insights into their effectiveness for our specific application.
- **Exploring model robustness against data drift and evolving medical terminology:** As the landscape of medical data continuously evolves, ensuring the long-term applicability of our models necessitates strategies to mitigate the impact of shifting data distributions and emerging terminologies. Investigating techniques such as continual learning, adaptive algorithms, and semantic drift detection can be considerable to maintain the performance and relevance of our models in dynamic healthcare environments.

CRediT authorship contribution statement

Berat Bozkurt: Writing – original draft, Methodology, Investigation, Formal analysis. **Kerem Coskun:** Writing – original draft, Methodology, Investigation, Formal analysis. **Gokhan Bakal:** Writing – review & editing, Supervision, Project administration, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The dataset we curated is available with four individual compact `DataFrame` format files and accessible via the link: https://shorturl.at/lotEV.

Acknowledgments

We appreciate the valuable comments and suggestions of reviewers.

Funding

No funds, grants, or other types of support were received.

References

[1] Arshia Rehman, Saeeda Naz, Imran Razzak, Leveraging big data analytics in healthcare enhancement: Trends, challenges and opportunities, Multimedia Syst. 28 (4) (2022) 1339–1371.
[2] Harsh Valecha, Aparna Varma, Ishita Khare, Aakash Sachdeva, Mukta Goyal, Prediction of consumer behaviour using random forest algorithm, in: 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering, UPCON, IEEE, 2018, pp. 1–6.
[3] Marcos Roberto Machado, Salma Karray, Ivaldo Tributino de Sousa, LightGBM: An effective decision tree gradient boosting method to predict customer loyalty in the finance industry, in: 2019 14th International Conference on Computer Science & Education, ICCSE, IEEE, 2019, pp. 1111–1116.
[4] Yanqing Zhang, Xuan Bi, Niansheng Tang, Annie Qu, Dynamic tensor recommender systems, J. Mach. Learn. Res. 22 (2021).
[5] Gokhan Bakal, Orhan Abar, On comparative classification of relevant COVID-19 tweets, in: 2021 6th International Conference on Computer Science and Engineering, UBMK, IEEE, 2021, pp. 287–291.
[6] Mehmet Umut Salur, Ilhan Aydin, A novel hybrid deep learning model for sentiment classification, IEEE Access 8 (2020) 58080–58093.
[7] Sulaf Elshaar, Samira Sadaoui, Semi-supervised classification of fraud data in commercial auctions, Appl. Artif. Intell. 34 (1) (2020) 47–63.
[8] Mohamed Elhoseny, K. Shankar, J. Uthayakumar, Intelligent diagnostic prediction and classification system for chronic kidney disease, Sci. Rep. 9 (1) (2019) 1–14.
[9] Gokhan Bakal, Ramakanth Kavuluru, Predicting treatment relations with semantic patterns over biomedical knowledge graphs, in: International Conference on Mining Intelligence and Knowledge Exploration, Springer, 2015, pp. 586–596.
[10] Wei-Hung Weng, Kavishwar B. Wagholikar, Alexa T. McCray, Peter Szolovits, Henry C. Chueh, Medical subdomain classification of clinical notes using a machine learning-based natural language processing approach, BMC Med. Inform. Decis. Mak. 17 (2017) 1–13.
[11] Fan Jiang, Carson K. Leung, Adam G.M. Pazdor, Big data mining of social networks for friend recommendation, in: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM, IEEE, 2016, pp. 921–922.
[12] Harold Borko, Myrna Bernick, Automatic document classification, J. ACM 10 (2) (1963) 151–162.
[13] Jiu-Zhen Liang, SVM multi-classifier and web document classification, in: Proceedings of 2004 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 04EX826), vol. 3, IEEE, 2004, pp. 1347–1351.
[14] Larry M. Manevitz, Malik Yousef, One-class SVMs for document classification, J. Mach. Learn. Res. 2 (Dec) (2001) 139–154.
[15] Sergio G. Burdisso, Marcelo Errecalde, Manuel Montes-y Gómez, A text classification framework for simple and effective early depression detection over social media streams, Expert Syst. Appl. 133 (2019) 182–197.
[16] Betul Erkantarci, Gokhan Bakal, An empirical study of sentiment analysis utilizing machine learning and deep learning algorithms, J. Comput. Soc. Sci. (2023) 1–17.
[17] Shuo Jiang, Jie Hu, Christopher L. Magee, Jianxi Luo, Deep learning for technical document classification, IEEE Trans. Eng. Manage. (2022).
[18] Bichitrananda Behera, G. Kumaravelan, Prem Kumar, Performance evaluation of deep learning algorithms in biomedical document classification, in: 2019 11th International Conference on Advanced Computing, ICoAC, IEEE, 2019, pp. 220–224.
[19] Alberto Blanco, Arantza Casillas, Alicia Pérez, Arantza Diaz de Ilarraza, Multi-label clinical document classification: Impact of label-density, Expert Syst. Appl. 138 (2019) 112835.
[20] Francesco Sovrano, Monica Palmirani, Fabio Vitali, Deep learning based multi-label text classification of UNGA resolutions, in: Proceedings of the 13th International Conference on Theory and Practice of Electronic Governance, 2020, pp. 686–695.
[21] NLM, PubMed, 2023, https://pubmed.ncbi.nlm.nih.gov.
[22] Trevor Hastie, Robert Tibshirani, Jerome Friedman, Trevor Hastie, Robert Tibshirani, Jerome Friedman, Overview of supervised learning, in: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer, 2009, pp. 9–41.
[23] Horace B. Barlow, Unsupervised learning, Neural Comput. 1 (3) (1989) 295–311.
[24] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, Anil A. Bharath, Generative adversarial networks: An overview, IEEE Signal Process. Mag. 35 (1) (2018) 53–65.
[25] David G. Kleinbaum, K. Dietz, M. Gail, Mitchel Klein, Mitchell Klein, Logistic Regression, Springer, 2002.
[26] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al., Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[27] Tianqi Chen, Carlos Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.
[28] Anna Veronika Dorogush, Vasily Ershov, Andrey Gulin, CatBoost: Gradient boosting with categorical features support, 2018, arXiv preprint arXiv:1810.11363.
[29] Aurélien Géron, Hands-on machine learning with scikit-learn, keras, and TensorFlow, " O'Reilly Media, Inc.", 2022.
[30] Haitao Wang, Jie He, Xiaohong Zhang, Shufen Liu, A short text classification method based on N-gram and CNN, Chin. J. Electron. 29 (2) (2020) 248–254.
[31] Yuandong Luan, Shaofu Lin, Research on text classification based on CNN and LSTM, in: 2019 IEEE International Conference on Artificial Intelligence and Computer Applications, ICAICA, IEEE, 2019, pp. 352–355.
[32] Rui Fu, Zuo Zhang, Li Li, Using LSTM and GRU neural network methods for traffic flow prediction, in: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation, YAC, IEEE, 2016, pp. 324–328.
[33] Cyril Goutte, Eric Gaussier, A probabilistic interpretation of precision, recall and F-score, with implication for evaluation, in: European Conference on Information Retrieval, Springer, 2005, pp. 345–359.
[34] Manoj P. Rai, Prabhjot S. Bedi, Edwin B. Marinas, Supratik Rayamajhi, Urinary bladder mass due to chronic lymphocytic leukaemia, Case Rep. 2018 (2018) bcr–2017.