Impact of Feature Types on Boundary Detection Between Metadata and Body in Persian Theses

Nima Shadman[*] Jalal A. Nasiri[†]

Abstract

This paper addresses the challenge of distinguishing header metadata from body content in Persian electronic theses and dissertations. Accurate classification of these sections aids tasks such as metadata extraction from scientific documents and plays a crucial role in increasing the efficiency and retrieval of information in digital libraries. Several machine learning models were employed to achieve this goal. Additionally, five distinct feature types were utilized: Heuristic, Sequential, Lexical, Formatting, and Geometric. The dataset consisted of nearly 230,000 paragraphs extracted from 106 Persian ETDs, with the metadata class representing only 8.6%. After preprocessing, Random Forest slightly outperformed SVM and Naïve Bayes. Moreover, our findings indicate that features of sequential type notably impact the classification metrics.

Keywords: Paragraph Classification, Metadata Extraction, Persian Scientific Documents, Features Fusion

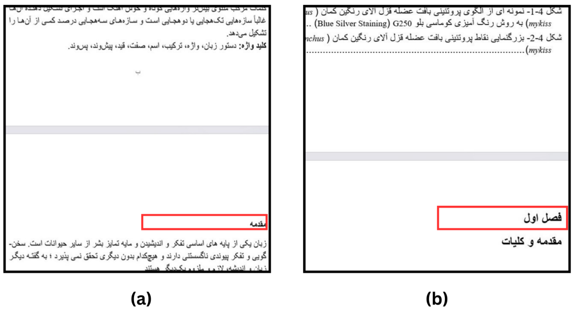
1 Introduction

In modern digital libraries, it is becoming increasingly difficult to extract metadata from documents due to the growing volume of scientific literature and their wide variety of layouts and styles [19]. Metadata is a type of data that provides information about other data. In scientific research, header metadata typically includes the title, author, abstract, keywords, and more. Processing the entire document to extract header metadata is inefficient, as they usually appear in the initial segments of documents. Therefore, detecting the boundary between header metadata and body content is crucial for efficient metadata extraction.

The majority of research in this area extracts metadata by relying on either rule-based methods, such as regular expressions, or machine learning-based methods. The latter combines natural language processing techniques with machine learning algorithms [13]. Most studies in this area have focused on English documents in PDF format, with only a few in Languages like Persian.

This paper aims to detect the boundary between header metadata and body content, specifically identifying the paragraph marking the start of the main body content, such as the 'First Chapter' or 'Introduction,' in Persian Electronic Theses and Dissertations (ETDs) in DOCX format. This paragraph usually appears after the table of contents, keywords, or abstract. Figure 1 shows examples of the boundary between header metadata and the main body.

To detect this boundary, we will incorporate various machine learning algorithms and compare their performance. Furthermore, we will leverage 5 different types of features and analyze their impact on the performance through combinations and fusion.



کلمات مرکب مثلوی بیش‌تر واژه‌هایی کوتاه و خوش آهنگ است و اجزای تشکیل دهنده آن‌ها
غالباً سازه‌هایی تک‌هجایی یا دوهجایی است و سازه‌های سه‌هجایی درصد کمی از آن‌ها را
تشکیل می‌دهد.
کلید واژه: دستور زبان، واژه، ترکیب، اسم، صفت، قید، پیش‌وند، پس‌وند.

ب

مقدمه

زبان یکی از پایه های اساسی تفکر و اندیشیدن و مایه تمایز بشر از سایر حیوانات است. سخن-
گویی و تفکر پیوندی ناگسستنی دارند و هیچ‌کدام بدون دیگری تحقق نمی پذیرد ؛ به گفته دیگر
زبان و اندیشه، لازم و ملزوم یکدیگر هستند

شکل 4-1- نمونه ای از الگوی پروتئینی بافت عضله قزل آلای رنگین کمان (*us*
mykiss) به روش رنگ آمیزی کوماسی بلو G250 (Blue Silver Staining) ..
شکل 4-2- بزرگنمایی نقاط پروتئینی بافت عضله قزل آلای رنگین کمان (*nchus*
mykiss)..

فصل اول

مقدمه و کلیات

Figure 1: Examples of Boundary Between Header Metadata and Main Body

In the following sections, we will provide an in-depth review of related works, describe our methodology and workflow, and analyze the influence of different classification algorithms and feature types on predictive performance.

2 Related Works

As discussed above, most studies employ rule-based or machine learning-based methods. In studies that take a rule-based approach, the components of the document are matched with predefined rules. For instance, in [2], the title of the paper is extracted from the PDF file using layout features, font and size characteristics of text, and other metadata are extracted utilizing fixed rule sets. [11] and [9] use a template matching-based method

[*] Natural Language Processing Lab, Ferdowsi University of Mashhad, nimashadman@alumni.um.ac.ir

[†] Department of Applied Mathematics, Ferdowsi University of Mashhad, jnasiri@um.ac.ir

to extract header metadata from PDF files. [10] describes a metadata extraction system for PDF files that uses knowledge-based methods for header metadata and predefined rules for reference information. Moreover, [14] focuses on using Regular Expressions to extract header metadata from PDF, DOCX, and text files.

Many studies adopt a machine learning approach. [7] proposes a Conditional Random Field (CRF) model, which combines text-based and visual features to extract metadata from cover pages of scanned ETDs. In [19], most steps are implemented using supervised and unsupervised machine learning techniques, such as Support Vector Machine (SVM) for initial and metadata zone classification, K-means for dividing the references zone into reference strings, and CRF for extracting metadata information from these reference strings. The proposed method in [4] focuses on detecting the references section and extracting the references metadata by using random forest for line classification as either reference or non-reference and then applying CRF to reference lines. [1] uses various classification algorithms such as SVM, K-Nearest Neighbors (K-NN), decision trees, and more for header metadata extraction from PDF files and concludes that SVM achieved the best results compared to the others. [12] employs deep learning networks to model image and text information of paper headers, combining convolutional neural networks and long short-term memory networks.

There is also research conducted on scientific documents in languages other than English. For example, [5] presents a method using Mask R-CNN that can also extract metadata from German publications. Additionally, [17] uses CRF to extract reference metadata from Korean research papers.

However, as pointed out earlier, there have not been many studies on metadata extraction from Persian scientific documents. In the initial stage of the method proposed in [15], the main body of the document is detected and discarded. After that, the remaining paragraphs are classified into metadata classes. Both stages in this method use SVM, with the bat algorithm employed to set its hyperparameters. [3] explores Ensemble approaches, combining different kinds of classifiers such as SVM, K-NN, and decision trees to extract metadata from Persian theses in DOCX format. Moreover, [18] initially detects the header and references sections of Persian research papers in PDF format, and then utilizes CRF to extract metadata.

3 Methodology

In this section, we will propose a method to detect the boundary between header metadata and body, labeling each paragraph as either metadata or body, enabling us to extract different header metadata without the need to process the main body. Figure 2 shows the 6 stages of this method. We will go through each stage in the following subsections.

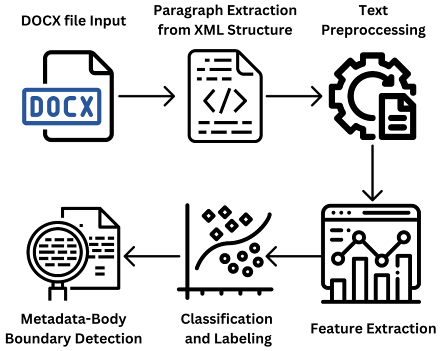

DOCX file Input
Paragraph Extraction from XML Structure
Text Preproccessing
DOCX
Metadata-Body Boundary Detection
Classification and Labeling
Feature Extraction

Figure 2: Overview of Boundary Detection Workflow

3.1 DOCX File Input

Since our data is in DOCX format, we can process the properties of each paragraph's text, including font size, font name, margins, and more. However, we cannot detect soft page breaks, so we are unable to determine which page each paragraph is on.

3.2 Paragraph Extraction from XML Structure

To avoid missing any text in the document, such as the text on shapes and in page footers, we will extract the paragraphs from the XML Structure of the document. In order to do this, we will look for <w:p> tags in the XML structure and extract the text and its properties.

3.3 Text Preprocessing

It is crucial to preprocess the text before proceeding, due to certain challenges in Persian text processing. For instance, some Persian letters have different Unicode representations in Arabic and Persian. Moreover, Persian numbers have a different Unicode representation than Arabic or Latin numbers.

Another challenge is related to the Zero Width Non-Joiner (ZWNJ) character. The ZWNJ character is used in Persian texts to create a space between two characters of a word without breaking them into separate words. composite words can be typed with the ZWNJ character instead of the space character.

There are also Arabic-derived characters in Persian such as Hamza, which may or may not appear in a specific word.

The issues mentioned are addressed by normalization in the preprocessing stage, where all undesired characters are removed and Arabic Unicode characters are replaced with their Persian counterparts. Additionally, all Latin numbers are replaced with Persian ones.

3.4 Feature Extraction

The features selected in this study are similar to features proposed in [19] and [15], with some modifications and additions. CERMINE [19] proposes 5 types of features: Heuristic, Sequential, Lexical, Formatting and Geometric features. Heuristic features are based on the text's nature. Sequential features depend on paragraph order. Lexical features are features related to specific keywords in the text. Formatting features describe the style of the paragraph, while Geometric features are derived from geometric properties.
This study proposes 14 features, as listed in Table 1. The first four heuristic features calculate the proportion of specific characters in the text by dividing the number of those characters by the total number of characters in the paragraph. Paragraph Relative Position is also calculated by dividing the paragraph index by the number of paragraphs in the document.

Feature	Type
Digit Frequency Ratio	Heuristic
Dot Frequency Ratio	Heuristic
Parentheses Frequency Ratio	Heuristic
Punctuation Frequency Ratio	Heuristic
Does This Paragraph Start with Digit?	Heuristic
Word Count	Heuristic
Paragraph Relative Position	Sequential
Prior Label	Sequential
Two-Step Prior Label	Sequential
Do First Chapter-Related Words Appear in Text?	Lexical
Do Header-Related Words Appear in Text?	Lexical
Font Size	Formatting
Width-Height Ratio	Geometric
Is This Paragraph Center Aligned?	Geometric

Table 1: Features For Metadata-Body Boundary Detection

First chapter-related words refer to words that typically appear in the first paragraph of the main body, such as Persian equivalents of the words 'introduction', 'chapter one', and so on. Header-related words include keywords that typically appear alongside header metadata, including Persian equivalents of the words 'title', 'author', 'abstract', etc.
Since soft breaks for pages and lines cannot be detected when processing a DOCX document, it is not possible to determine how many lines a paragraph has. However, we can approximate this using information such as font size and margins. Using this approximation and the line spacing information, we can calculate the height of a paragraph. We can also calculate the width of a paragraph from page width, and left and right margins.

3.5 Classification and Labeling

Before applying the machine learning models to the dataset, the data points go through a Normalization process. The Min-Max Scaler is used for Normalization, which scales data to the range (0,1) by transforming each feature value based on the minimum and maximum value of that feature [8].
We will utilize these machine learning algorithms, with an explanation of each included in the subsequent parts of this subsection:

- Support Vector Machine (RBF kernel, $C = 16$, $gamma = 2^{-4}$)
- Random Forest (number of trees = 100, maximum depth = $None$)
- Naïve Bayes (Gaussian Naïve Bayes)

The hyperparameters for these models were selected by conducting a Grid Search, considering both performance results and training time constraints.

3.5.1 Support Vector Machine

Support Vector Machine (SVM) is a machine learning algorithm for binary classification problems [16]. SVM focuses on finding the optimal hyperplane that has the most distance from the data points of both classes. The parameters for this optimal hyperplane can be obtained by solving the following optimization problem:

$$\min_{w} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i \tag{1}$$

$$s.t.\ \ y_i(w^T x_i + b) + \xi_i \geq 1,\ \forall i$$

Where ξ_i is the slack variable associated with x_i sample and C is the penalty parameter. It should be noted that the optimization problem in (1) can be solved if the two classes are linearly separable. In case the data is not linearly separable, it is transformed into a higher-dimensional space to achieve linear separability in the new space.

3.5.2 Random Forest

Random Forest is a combination of tree predictors such that each tree depends on the values of a random vector sampled independently [6]. This means that for the kth tree, a random vector Θ_k is generated that is independent of random vectors $\Theta_1, \ldots, \Theta_{k-1}$ but with the same distribution. A tree is grown using the training set and Θ_k, forming a classifier $h(x, \Theta_k)$, where x is an input vector. After an abundance of trees is generated, they vote for the most popular class.

3.5.3 Naive Bayes

The Naïve Bayes classification algorithm is based on the Bayes rule, which assumes that the features are all conditionally independent [20]. Let X be the set of n attributes which are conditionally independent of one another given Y. Hence, we have:

$$P(X_1,\ldots,X_n|Y)=\prod_{i=1}^{n}P(X_i|Y) \tag{2}$$

To derive the Naïve Bayes algorithm, we assume that Y is any discrete-valued variable and the attributes in X are any discrete or real-valued attributes. The probability that Y will take on its kth possible value according to Bayes' rule, can be expressed under the assumption of conditional independence of the X_i given Y as follows:

$$P(Y=y_i|X_1\ldots X_n)=\frac{P(Y=y_k)\prod_i P(X_i|Y=y_k)}{\sum_j P(Y=y_j)\prod_i P(X_i|Y=y_j)} \tag{3}$$

3.6 Metadata-Body Boundary Detection

When all paragraphs are labeled as either metadata or body, we start from the first paragraph and proceed, until there is a sequence of consecutive paragraphs labeled as body, whose count exceeds a specific threshold. Upon finding that, we define the Metadata-Body boundary as the first paragraph of this sequence. If the threshold is not met, we find the longest sequence of consecutive paragraphs predicted as the main body paragraph and set their first paragraph as the boundary.

4 Evaluation

This section will first overview the dataset, followed by a comparison of each model's results and an evaluation of feature type impacts.

4.1 Dataset

For evaluation of the proposed method, the dataset consists of 106 Persian ETDs in DOCX format. A total of 228,321 paragraphs were extracted from these documents, with 208,704 paragraphs in the body class (91.4%) and 19,617 paragraphs in the metadata class (8.6%). As it is evident, the dataset is imbalanced, since scientific documents typically contain significantly more body paragraphs than metadata ones. Moreover, the scientific documents are categorized into 4 groups based on their field of study. Figure 3 illustrates the distribution of each category using a pie chart.

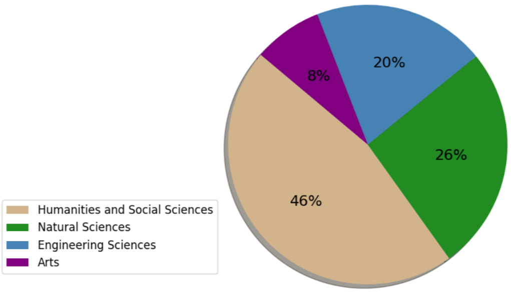


Figure 3: Categorization of Scientific Documents by Field of Study

4.2 Performance Evaluation

In this paper, we consider accuracy, precision, recall, and F1-score metrics for performance evaluation. The evaluation was performed using 10-fold cross-validation. This means that the dataset was divided into 10 subsets, with the model trained 10 times, each time using one subset for testing and the others for training. Therefore, the values provided in this section are the averages of these 10 models for each machine learning algorithm. Since the dataset is imbalanced, we use macro averaging instead of weighted averaging for precision, recall, and F1-score metrics, to balance the impact of performance on both majority and minority classes. As mentioned in the previous section, three classification algorithms are employed for this task. Table 2 presents the performance of each algorithm in 10-fold cross-validation. As shown in Table 2, Random Forest slightly outperforms the other models in classification metrics

	Random Forest	SVM	Naïve Bayes
Average Accuracy	**99.95%**	99.92%	99.84%
Average Precision	**99.78%**	99.59%	99.10%
Average Recall	**99.93%**	99.92%	99.91%
Average F1-score	**99.85%**	99.76%	99.50%

Table 2: Performance of Random Forest, SVM, and Naïve Bayes in 10-fold

Table 3 and Figure 4 show a detailed classification report and the cumulative confusion matrix for the Random Forest model, respectively. As expected, the model's performance on the majority class is exceptional, with all metrics above 99.95%. On the other hand, the model performs impressively on the minority class as well, achieving an average F1-score of 99.73%.

As mentioned earlier, we want to analyze the impact of each feature type and compare their effectiveness on performance. Table 4 provides each model's performance in terms of macro average F1-score across

	Average Precision	Average Recall	Average F1-score
Metadata	99.54%	99.91%	99.73%
Body	99.99%	99.96%	99.97%

Table 3: Classification Report for the Random Forest Model

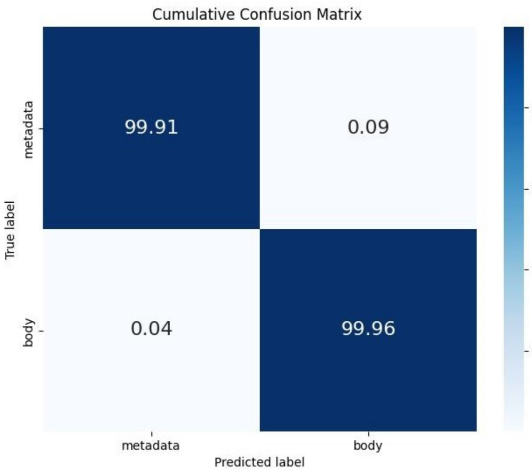


Figure 4: Cumulative Confusion Matrix of the Random Forest Model

feature types fusion. Heu represents Heuristic features, while Seq, Lex, For, and Geo represent Sequential, Lexical, Formatting, and Geometric features, respectively. As seen in Table 4, Sequential features have the most influence on all model performances. Even to the extent that each model tends to perform better when trained only on the 3 sequential features, compared to when trained with all other 11 features combined. It can also be noted that Random Forest considerably outperforms other models when sequential features are absent from the feature set. Furthermore, the results show that Heuristic, Lexical, Geometric, and Formatting features do not significantly impact the performance of each model when any one of them is absent from the feature set.

The SVM model tends to perform nearly in the same way when Formatting, Heuristic, or Geometric features are excluded. Additionally, the Naïve Bayes model has a slightly better performance when trained only on sequential and lexical features. It is also evident in Table 4 that the Random Forest model does not outperform the other two models in all cases. For instance, it is marginally outperformed by the other two models when trained only on sequential features.

Feature Set	**Random Forest**	**Support Vector Machine**	**Naïve Bayes**
Heu	82.06%	57.47%	54.73%
Seq	99.45%	99.71%	**99.56%**
Seq + Lex	99.51%	99.71%	**99.56%**
Seq + Geo	99.63%	99.71%	99.53%
Lex + Geo	75.59%	50.21%	49.88%
Seq + Heu	99.62%	99.71%	99.52%
Heu + Lex	83.38%	58.34%	55.80%
Heu + Geo	86.65%	59.61%	57.19%
Heu + Lex + Geo	87.38%	60.28%	57.23%
Seq + Lex + Geo	99.69%	99.75%	99.53%
Heu + Seq + Lex	99.73%	99.71%	99.52%
Heu + Seq + Geo	99.68%	99.71%	99.50%
All without For	99.80%	99.75%	99.51%
All without Heu	99.79%	**99.76%**	99.52%
All without Seq	90.15%	60.32%	57.59%
All without Geo	99.82%	99.75%	99.52%
All without Lex	99.79%	99.71%	99.50%
All	**99.85%**	**99.76%**	99.50%

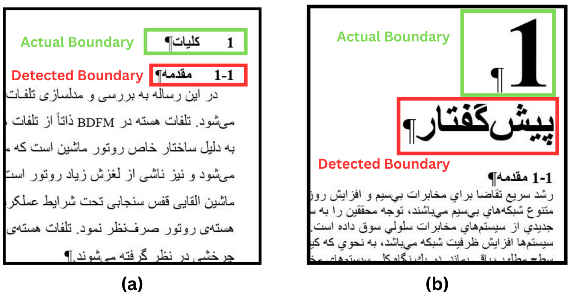
Table 4: Macro Average F1-score for each Model over Features Fusion

Table 5 evaluates the effectiveness of our boundary detection method using the Random Forest model on new, unseen data. To do this, the trained random forest model was tested on 50 new ETDs to detect the Metadata-Body boundary in each document. The Strict Accuracy metric refers to the proportion of correctly detected boundaries where the predicted boundary index matches the actual boundary index exactly. Relaxed Accuracy, on the other hand, allows a deviation of up to 5 paragraph indices. This means that if the detected boundary deviates by fewer than 5 paragraph indices from the actual boundary, it is still considered a correct detection. Figure 5 illustrates examples where the detected boundary deviates from the actual boundary by a margin of just one paragraph index. Average Deviation represents the average number of indices between the detected and actual boundary. Moreover, Average Metadata Paragraphs Missed shows the number of metadata paragraphs that are incorrectly placed after the detected boundary across the new ETDs.

As shown in Table 5, the model performs well, detecting the exact boundary in 64% of the new ETDs, which increases to 84% when a small margin of error is allowed. The boundary detected by this model deviates by an average of 6 paragraphs from the actual boundary and misses less than 2 metadata paragraphs on average

Metric	**Value**
Strict Accuracy	64%
Relaxed Accuracy	84%
Average Deviation	6.06
Average Metadata Paragraphs Missed	1.78

Table 5: Evaluation of Boundary Detection Effectiveness on New ETDs Using Random Forest



1 كليات¶
1-1 مقدمه¶
در این رساله به بررسی و مدلسازی تلفات
می‌شود. تلفات هسته در BDFM ذاتاً از تلفات
به دلیل ساختار خاص روتور ماشین است که
می‌شود و نیز ناشی از لغزش زیاد روتور است
ماشین القایی قفس سنجابی تحت شرایط عملکر
هسته‌ی روتور صرف‌نظر نمود. تلفات هسته‌ی
چرخشی در نظر گرفته می‌شوند.¶

1¶
پیش‌گفتار¶
1-1 مقدمه¶
رشد سريع تقاضا براي مخابرات بي‌سيم و افزايش روز
متنوع شبكه‌هاي بي‌سيم مي‌باشند، توجه محققين را به س
جديدي از سيستم‌هاي مخابرات سلولي سوق داده است.
سيستم‌ها افزايش ظرفيت شبكه مي‌باشد، به نحوي كه كي

Figure 5: Examples of Detected Boundaries with Small Deviation from the Actual Boundary

when discarding the content after the detected boundary.

Table 6 compares the average F1-score of our method with the proposed method in [19] for zone classification in scientific publications and the method suggested by the study in [15] to detect Metadata-Body boundary in Persian theses. The comparison shows that our method achieves a better average F1-score than both methods in the mentioned studies.

Method	Average F1-Score
Proposed Method	**99.8%**
(Rahnama et al. 2020) [15]	95.6%
(Tkaczyk et al. 2015) [19]	93.9%

Table 6: Average F1-Score Comparison with Related Studies for Boundary Detection

5 Conclusion

In this paper, we presented a method to detect the boundary between the metadata and the main body of Persian ETDs in DOCX format, of which metadata extraction systems can take advantage to increase efficiency. A total of 14 features across 5 types are introduced for this method, and several machine learning algorithms were utilized for this task. The results show that sequential features have the most impact on performance, and the random forest model has a better performance than the other models mentioned in this paper. The boundary detection method outperformed similar techniques in metadata extraction studies.

Future studies can focus on detecting sections such as table of contents or bibliography, as well as extracting metadata from scanned Persian ETDs.

References

[1] M. W. Ahmed and M. T. Afzal. Flag-pdfe: Features oriented metadata extraction framework for scientific publications. *IEEE Access*, 8:99458–99469, 2020.

[2] J. Azimjonov and J. Alikhanov. Rule based metadata extraction framework from academic articles. *ArXiv*, abs/1807.09009, 2018.

[3] E. Beydaghi, M. Rahnama, and J. A. Nasiri. Ensemble approach for metadata extraction in persian theses. *2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, pages 1–5, 2020.

[4] Z. Boukhers, S. Ambhore, and S. Staab. An end-to-end approach for extracting and segmenting high-variance references from pdf documents. *2019 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 186–195, 2019.

[5] Z. Boukhers, N. Beili, T. Hartmann, P. Goswami, and M. A. Zafar. Mexpub: Deep transfer learning for metadata extraction from german publications. *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 250–253, 2021.

[6] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

[7] M. H. Choudhury, H. R. Jayanetti, J. Wu, W. A. Ingram, and E. A. Fox. Automatic metadata extraction incorporating visual features from scanned electronic theses and dissertations. *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 230–233, 2021.

[8] B. Deepa and K. Ramesh. Epileptic seizure detection using deep learning through min max scaler normalization. *International journal of health sciences*, 2022.

[9] P. Flynn, L. Zhou, K. Maly, S. J. Zeil, and M. Zubair. Automated template-based metadata extraction architecture. In *International Conference on Asian Digital Libraries*, 2007.

[10] Z. Guo and H. Jin. Reference metadata extraction from scientific papers. *2011 12th International Conference on Parallel and Distributed Computing, Applications and Technologies*, pages 45–49, 2011.

[11] Z. Huang, H. Jin, P. Yuan, and Z. Han. Header metadata extraction from semi-structured documents using template matching. In *OTM Workshops*, 2006.

[12] R. Liu, L. Gao, D. An, Z. Jiang, and Z. Tang. Automatic document metadata extraction based on deep networks. In *Natural Language Processing and Chinese Computing*, 2017.

[13] Z. Nasar, S. W. Jaffry, and M. K. Malik. Information extraction from scientific articles: a survey. *Scientometrics*, 117:1931 – 1990, 2018.

[14] B. A. Ojokoh, O. S. Adewale, and S. O. Falaki. Automated document metadata extraction. *Journal of Information Science*, 35:563 – 570, 2009.

[15] M. Rahnama, S. M. H. Hasheminejad, and J. A. Nasiri. Automatic metadata extraction from iranian theses and dissertations. *2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, pages 1–5, 2020.

[16] M. S. Refahi, A. M. Mir, and J. A. Nasiri. A novel fusion based on the evolutionary features for protein fold recognition using support vector machines. *Scientific Reports*, 10, 2019.

[17] J.-W. Seol, W. Choi, H.-S. Jeong, H. Hwang, and H.-M. Yoon. Reference metadata extraction from korean research papers. In *International Conference on Mining Intelligence and Knowledge Exploration*, 2018.

[18] A. Tansazan and M. a. Mahdavi. Metadata extraction from persian scientific papers using crf model. *Library and Information Science Research*, 7(1):304–321, 2017.

[19] D. Tkaczyk, P. Szostek, M. Fedoryszak, P. J. Dendek, and L. Bolikowski. Cermine: automatic extraction of structured metadata from scientific literature. *International Journal on Document Analysis and Recognition (IJDAR)*, 18:317 – 335, 2015.

[20] Vikramkumar, B. Vijaykumar, and Trilochan. Bayes and naive bayes classifier. 2014.