Innovative Informatics and Artificial Intelligence Research (IIAIR)
Vol. 2, Issue 1, pp. 30–37, 2026
Received 4 Feb 2026; accepted 25 Apr 2026; published 30 Apr 2026
https://doi.org/10.35718/iiair.v2i1.8481931

Dom-Tree Based Automatic Classification Of Predatory Journals Using Doc2vec And Automated Machine Learning

Ersan Karimi[1], Gusti Ahmad Fanshuri Alfarisy[1], Bowo Nugroho[1], and Ahmad Fathan Hidayatullah[2]

[1]Department of Informatics, Institut Teknologi Kalimantan, Balikpapan, Indonesia
[2]Department of Informatics, Universitas Islam Indonesia, Indonesia

Corresponding author: Gusti Ahmad Fanshuri Alfarisy (gusti.alfarisy@lecturer.itk.ac.id)

To cite this article: E. Karimi, G. A. F. Alfarisy, B. Nugroho, A. F. Hidayatullah, "Dom-Tree Based Automatic Classification Of Predatory Journals Using Doc2vec And Automated Machine Learning," *Innovative Informatics and Artificial Intelligence Research*, vol. 2, issue 1, 2026. [Online]. Available: https://doi.org/10.35718/iiair.v2i1.8481931

Gusti Ahmad Fanshuri Alfarisy serves as an Editor of IIAIR but was not involved in the peer-review process of this article

Abstract

Predatory journals threaten academic integrity by offering publication without proper peer review. Indonesia ranked second globally, with 16.73% of articles suspected to have been published in predatory journals during 2015–2017. This study aims to develop a method for classifying the web pages of predatory journals using a combination of Distributed Representations of Documents (Doc2Vec) and Automated Machine Learning (AutoML) based on the structure of the Document Object Model (DOM) tree. The dataset of predatory journals was collected from Kaggle, while non-predatory journals were obtained from the Directory of Open Access Journals (DOAJ). The main pages of journal websites were collected through web scraping and converted into a DOM corpus using two traversal approaches: Depth-First Search (DFS) and Breadth-First Search (BFS). The DOM corpus was then vectorized using Doc2Vec and automatically classified with AutoML from Auto-Sklearn. The evaluation was conducted using accuracy and macro avg F1-score metrics for each traversal method and training time configuration. AutoML training was tested within a range of 15 to 120 minutes, in 15-minute intervals. The best model for BFS was obtained at 15 minutes of training with a macro avg F1-score of 0.7812 and an accuracy of 0.9196. Meanwhile, the best model for DFS was achieved at 90 minutes of training with a macro avg F1-score of 0.7853 and an accuracy of 0.9255. These results indicate that the traversal method used to construct the DOM corpus influences the performance of the predatory journal classification model. DFS tends to yield better performance than BFS in the context of Doc2Vec and AutoML based on the DOM tree structure, as reflected in both accuracy and macro avg F1-score.

Keywords: predatory journals; machine learning; classification; automated machine learning; deep learning

1. Introduction

In recent decades, the advancement of digital technology and the emergence of open access (OA) publishing have significantly transformed the academic publishing landscape [1]. These developments have made scientific literature more accessible than ever before [2], removing barriers to knowledge and enabling scholars worldwide particularly in developing countries to access and contribute to the global research community without the paywalls of traditional publishing. Open access, in principle, enhances transparency, reproducibility, and equity in science by allowing anyone to read, download, and distribute scholarly work without cost [3]. However, this shift has not been without consequences. Alongside reputable OA journals, a parallel and more insidious phenomenon has emerged: predatory publishing.

Predatory journals are entities that exploit the open-access model unethically [4], [5]. This phenomenon has intensified alongside the rapid growth of open-access publishing over the past two decades[1], during which increasing publication pressure in academia has been systematically exploited by dubious publishers [6][7]. They charge publication fees to authors without providing the essential academic services expected of legitimate publishers, most notably rigorous peer review, transparent editorial processes, and scholarly curation [1], [2]. These journals often mislead authors with fabricated impact factors, inclusion in fake indexes, and promises of rapid publication, thus compromising the integrity of academic dissemination. Despite attempts to mimic reputable journals, their poor editorial standards and low-quality presentation often raise doubts upon closer inspection [8]. While the problem affects the global academic community, researchers in regions with limited institutional support or strong publication pressure such as Southeast Asia, South Asia, and parts of Africa are particularly vulnerable.

Indonesia, in particular, has become a concerning hotspot in

[1]Berlin Declaration on Open Access to Knowledge in the Sciences and Humanities (https://openaccess.mpg.de/Berlin-Declaration)

the global discussion around predatory publishing. A bibliometric study conducted by Macháček and Srholec [9] revealed that from 2015 to 2017, approximately 16.73% of 164,000 articles published by Indonesian scholars were likely published in predatory journals placing Indonesia second globally in this regard. These alarming numbers underline the urgency for scalable and systematic solutions to combat academic fraud, safeguard the reputation of legitimate scholars, and maintain the credibility of Indonesia's growing research output. Unfortunately, the identification and detection of predatory journals is a complex task, especially when done manually.

Traditional detection methods often rely on human judgment, blacklists (such as Beall's List), and community-maintained whitelists. However, these approaches are susceptible to several limitations: subjectivity, incomplete coverage, infrequent updates, and potential bias against new but legitimate open-access journals [10, 11]. Moreover, the increasing professionalization of predatory journal websites which often mimic legitimate academic portals in layout, language, and claims of peer review has rendered visual inspection and conventional content-checking unreliable [12, 13]. As such, there is a growing consensus among researchers that automated methods are required to ensure consistent, scalable, and objective evaluation of journal credibility.

In response to this need, recent studies have proposed a variety of machine learning (ML) and natural language processing (NLP) techniques to automatically detect predatory journals based on different data sources. Some approaches focus on textual and metadata features, such as journal titles, publisher names, editorial board members, submission-to-publication time, and declared policies. For instance, Chen et al. [14] used TF-IDF and Bag-of-Words (BoW) features with a Random Forest classifier and achieved an F1-score of 0.98. Adnan et al. [15] built a model based on heuristics-based indicators, including impact factor inconsistencies and scope mismatches, and achieved up to 0.98 accuracy using Support Vector Machines (SVMs). These studies suggest that metadata-based features can offer strong signals, especially when curated and standardized.

Advancements in deep learning have also introduced the use of neural networks for this task. Ateeq and Al-Khalifa [16] developed an intelligent framework using Convolutional Neural Networks (CNNs) on text-based inputs, obtaining F1-scores up to 0.96. Another work by Sharma et al. [17] proposed a big-data–driven AI system that combines journal and publisher information from multiple legitimate and illegitimate databases. The system employs LSTM-based text classification together with graph-informed features to categorize venues as predatory, suspicious, or non-predatory. Experiments on approximately 50,000 records reported an accuracy of 94%. However, the approach requires substantial computational resources and is sensitive to data quality and labeling consistency, which may limit its scalability.

Transformer-based models like ALBERT and LSTM architectures have also been explored to handle complex semantic representations of journal descriptions. However, these models typically require large, labeled datasets and are highly dependent on the availability and quality of the input data. Moreover, metadata is not always reliable, and in many cases, it is manipulated or fabricated by predatory publishers to deceive detection systems. Another concern with these methods is interpretability and reliability. Systems such as the Academic Journal Predatory Checking (AJPC) tool have come under scrutiny for high false-positive rates and lack of transparency. Teixeira da Silva [18] criticized such systems for misclassifying over 17,000 journals, many of which were actually legitimate, thereby questioning the reliability of black-box machine learning tools. To counter this, newer studies have adopted interpretable ML frameworks, such as SHAP (SHapley Additive exPlanations), to explain classification decisions and increase trust in automated systems [19].

Despite the sophistication of these approaches, a common dependency among them is their reliance on accessible, structured, and accurate metadata or textual content, a requirement that is often unmet in the wild. This creates a gap for methods that can operate independently of content semantics, and instead, focus on alternative cues such as page layout, structure, and organization. This is where Document Object Model (DOM) analysis becomes relevant.

The DOM is the structured representation of an HTML document, often modeled as a tree where each node represents an element in the web page. DOM analysis has shown promise in other domains such as phishing detection, where researchers have used DOM traversal and structural embedding to distinguish between benign and malicious web pages [20]. However, its application in identifying predatory journal websites remains limited.

This study aims to address this gap by introducing a novel method to classify predatory journals using DOM-based structural analysis. Specifically, we apply two well-known traversal algorithms, namely Depth-First Search (DFS) and Breadth-First Search (BFS) to linearize the DOM tree of each journal homepage into a textual sequence of HTML tags. This sequence, referred to as the DOM corpus, is then vectorized using Doc2Vec, a neural model that captures semantic similarities between sequences of tokens [21]. The resulting vectors are used as input for an automated machine learning (AutoML) process using Auto-Sklearn [11], which automates model selection, hyperparameter optimization, and validation.

The main objective of this study is to evaluate the effectiveness of DOM traversal methods (DFS and BFS) in extracting structural features for classifying journal websites as predatory or non-predatory. We hypothesize that while content can be faked, structure often reveals underlying patterns, such as excessive use of tags, repetitive submission buttons, or shallow hierarchy depth, that are more difficult to mask. DOM-based methods are also language-agnostic, universally applicable to HTML websites, and resilient to textual obfuscation, making them highly suitable for this task.

In summary, this study contributes to the field by (1) applying DOM tree traversal for predatory journal detection which is a rarely explored area, (2) integrating Doc2Vec and AutoML for scalable and automated classification, and (3) benchmarking the effectiveness of structural-based approaches against conventional expectations. This approach offers a complementary pathway to traditional metadata analysis and enhances the robustness of automated journal screening systems.

2. Related Works

Research on the automatic classification and detection of predatory journals has gained increasing attention over the past decade as the threat posed by such journals becomes more prominent in global academic discourse. The fundamental goal of these studies is to develop reliable systems that can distinguish legitimate academic journals from predatory ones, using various features extracted from websites or related metadata. The growing number of submissions to questionable journals, along with the sophistication of predatory publishing practices, has made traditional manual detection methods insufficient, thereby motivating the adoption of machine learning (ML) and data-driven approaches.

In the early phases of this research field, most studies employed traditional machine learning algorithms trained on textual features derived from journal websites. These features often included data extracted from journal descriptions, aims and scope sections, editorial board listings, and author guidelines. The text data was vectorized using classic techniques such as Term Frequency-Inverse Document Frequency (TF-IDF) and Bag-of-Words (BoW), both of which transform natural language into structured numeric input for algorithms like Naive Bayes, Logistic Regression, and Random Forests. Chen et al. [14], for example, used a Random Forest classifier trained on TF-IDF vectors to classify journals as predatory or legitimate, reporting an F1-score as high as 0.98. These approaches were favored for their simplicity, interpretability, and relatively low computational cost.

However, reliance on textual content introduces inherent limitations. Textual information found on journal websites can vary significantly in quality, language, and format, especially in the context of international journals. Furthermore, predatory publishers are known to deliberately copy legitimate text from reputable journals, further diminishing the discriminative power of surface-level content analysis. The effectiveness of such models is also constrained by the necessity for well-structured and clean datasets, which are difficult to obtain in real-world settings where websites are poorly maintained or dynamically generated. These drawbacks pushed researchers to explore alternative features beyond surface text.

Subsequent studies began incorporating heuristic-based features hand-crafted indicators reflecting behavioral and structural patterns of predatory journals. Such heuristics include the time interval between manuscript submission and acceptance (often unrealistically short in predatory journals), frequency of publication, number of published volumes, editorial board completeness, fake indexing claims, and even the lexical patterns in editor email domains. Adnan et al. [15] demonstrated the potential of such indicators by developing a Support Vector Machine (SVM) classifier trained on metadata and heuristics, achieving an accuracy of 0.98. These features, while more robust than raw text, still require careful domain knowledge and extensive preprocessing. Additionally, they may be inconsistently available or deliberately obscured by predatory publishers.

To address the limitations of manual feature engineering, researchers turned toward deep learning techniques. These models, particularly Convolutional Neural Networks (CNNs) and recurrent neural networks like Long Short-Term Memory (LSTM), offer the ability to learn high-dimensional representations directly from raw data. Ateeq and Al-Khalifa [16] applied CNNs to journal metadata and achieved an F1-score of 0.96, showing improved robustness and generalization compared to classical models. Transformer-based architectures such as ALBERT have also been employed for modeling contextual dependencies within longer pieces of journal text. Deep learning models eliminate the need for manual feature extraction but require large volumes of labeled data and are computationally intensive. Moreover, their black-box nature makes them challenging to interpret, especially when deployed in sensitive academic or policy environments where explanations for decisions are necessary.

Parallel to the performance advancements in classification accuracy, concerns have arisen regarding the interpretability and transparency of these systems. Tools like the Academic Journal Predatory Checking (AJPC) system have been criticized for misclassifying reputable journals as predatory, raising ethical concerns about false positives and damage to the reputation of legitimate publishers. Teixeira da Silva [18] emphasized the risk of relying on opaque systems that cannot explain their decisions or be independently audited. This critique has led to the adoption of explainable AI (XAI) techniques in recent works. Wu et al. [19], for example, incorporated SHapley Additive exPlanations (SHAP) into an ensemble learning model to reveal which features influenced each classification outcome. The ability to provide transparent decision-making is crucial for gaining the trust of researchers, institutions, and journal editors who rely on such systems.

Despite the evolution of these methodologies, a common limitation persists: the dependence on accurate and accessible metadata or well-structured text content, which is often unavailable or manipulated. Predatory journals frequently falsify their editorial boards, misrepresent indexing status, and use legitimate-sounding names to deceive readers. The dynamic and deceptive nature of these journals makes metadata- and content-based approaches vulnerable. Additionally, not all journal platforms provide structured metadata through standardized formats such as DOAJ APIs or CrossRef endpoints, hindering scalability.

In light of these challenges, a relatively underexplored direction is the use of structural features of the web page itself. This approach shifts the focus from what is being said (text content) to how it is presented (HTML structure). Web pages are structured using the Document Object Model (DOM), which represents the hierarchical layout of elements such as , <p>, <a>, and <table>. While often disregarded in natural language processing tasks, this structure can encode implicit cues about the quality and intent of a website. For example, legitimate journals tend to follow standard publishing layouts with clear navigation, identifiable editorial information, and distinct article sections. In contrast, predatory journals may exhibit inconsistent structure, excessive use of hyperlinks, poor nesting of tags, and duplicated layouts across multiple journals.

A promising study by Feng et al. [20] explored DOM-based analysis in a different domain: phishing detection. They proposed a model that transformed the DOM tree of a phishing web page into a sequence of tags using a traversal algorithm, which was then vectorized using Doc2Vec, a method for learning distributed representations of documents. Their

results showed that even without textual content, the structure of the web page alone could be used effectively to classify malicious websites. This concept provides a compelling foundation for adapting DOM-based methods to the detection of predatory journals.

Building upon this idea, our research introduces a novel approach that applies DOM tree traversal, Doc2Vec embedding, and automated machine learning (AutoML) for classifying journal websites. The core insight is that while surface-level metadata and content can be forged, the underlying HTML structure tends to reflect the quality and consistency of the publisher's technical implementation. To extract meaningful patterns from this structure, we employ two traversal strategies, Depth-First Search (DFS) and Breadth-First Search (BFS) to flatten the DOM tree into a sequential representation. This sequence, which we term the DOM corpus, captures the hierarchical and syntactic layout of the journal homepage.

Each DOM corpus is embedded using Doc2Vec, which creates vector representations that preserve semantic and positional information between HTML tags. This process enables the model to learn patterns such as excessive nesting, repetition of submission links, absence of editorial layout, or anomalies in content structure. The embedded vectors are then classified using AutoSklearn, a leading AutoML library that automates model selection, hyperparameter optimization, and validation. This pipeline eliminates the need for manual feature engineering and allows the system to adaptively find the best model configuration based on the DOM-derived data.

The use of DOM-based analysis provides several significant advantages. First, HTML structure is universally available across websites and does not depend on linguistic content, making the approach language-independent and applicable globally. Second, the structure is harder to fake consistently compared to textual content or metadata, especially when websites are generated using cheap templates or lack technical sophistication. Third, DOM traversal avoids reliance on external databases, blacklists, or API connections, ensuring greater autonomy and scalability of the system.

In sum, while prior works have established strong foundations for predatory journal detection using text, heuristics, and deep learning, they remain constrained by the accessibility, integrity, and stability of content and metadata. Our work introduces a structural perspective that focuses on the presentation layer of journal websites, an area that has remained largely untapped. Through the integration of DOM traversal, Doc2Vec, and AutoML, our approach offers a complementary and robust method for automated classification of predatory journals, particularly in scenarios where conventional signals are unreliable or unavailable. This contribution not only advances the state of the art but also opens new avenues for research into the structural analysis of deceptive digital platforms.

3. Methods

The research began with problem identification and literature review regarding DOM tree traversal, Doc2Vec, and AutoML techniques. It proceeded with data collection from predatory and non-predatory journal websites, followed by preprocessing, vectorization, model training with AutoML, and evaluation using performance metrics.

3.1. Doc2Vec

Doc2Vec is an extension of Word2Vec designed to learn vector representations not only for words, but also for larger text units such as sentences, paragraphs, and documents [21]. Thus, each document can be encoded as a fixed-length vector that captures contextual and semantic information (or structural patterns) beyond simple word-frequency statistics. In general, Doc2Vec provides two main architectures: *Distributed Memory* (DM) and *Distributed Bag of Words* (DBOW) [21]. In this study, the DM architecture is adopted because it leverages the token-order context to produce more informative document representations.

In the DM scheme, the model learns a document vector (paragraph/document embedding) that is combined with the vectors of words in the context of the surrounding context to predict a target word within a sequence [21]. The document vector acts as a "memory" that stores the global topic/context of the document during training, allowing documents with similar characteristics to be mapped to nearby regions in the embedding space. With this mechanism, the resulting representation captures not only the tokens that appear but also the local context defined by a sliding window. An illustration of the *Distributed Memory* mechanism is shown in Fig. 1.

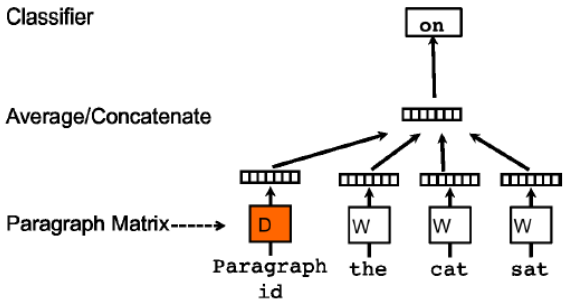

Figure 1: Illustration of the Doc2Vec Distributed Memory [21].

In this work, each journal web page is represented as a *DOM corpus*, i.e., a sequence of HTML tag tokens obtained from traversing the page's DOM structure (e.g., `<div>`, `<p>`, `<a>`, etc.) extracted from the `<body>` element. This tag sequence is treated as a "document" and is used to train Doc2Vec to produce a document-level embedding as a numerical feature vector.

3.2. Automated Machine Learning (AutoML): Auto-Sklearn

Automated Machine Learning (AutoML) is an approach that aims to automate key stages in the machine learning workflow, so that model development does not rely entirely on manual algorithm selection and hyperparameter tuning. In practice, AutoML searches for an appropriate pipeline configuration (e.g., preprocessing, model choice, and hyperparameter optimization) based on the characteristics of the data, with the goal of obtaining strong and reproducible performance efficiently.

In this study, we employ *Auto-Sklearn* [22][23], an AutoML framework built on top of *Scikit-Learn*. Auto-Sklearn automatically selects suitable learning algorithms for a given dataset and optimizes their hyperparameters, allowing practitioners to focus more on problem formulation and result analysis rather than extensive *trial-and-error* configuration [11].

Furthermore, Auto-Sklearn incorporates *meta-learning* to accelerate the search process by leveraging knowledge from previous optimization runs on earlier datasets [11]. In this phase, Auto-Sklearn extracts dataset meta-features (e.g., the number of instances and features) and performs a warm-start using high-performing configurations stored in its prior knowledge base.

After meta-learning, Auto-Sklearn conducts model and hyperparameter optimization (commonly via Bayesian optimization) to explore the configuration space more efficiently than naive random search. The final output is not limited to a single best model; instead, Auto-Sklearn can construct an *ensemble* of multiple top-performing models. This ensemble strategy combines predictions from several models to achieve more stable and accurate results and to improve generalization [11]. These properties make Auto-Sklearn well-suited for this work, as it automates model selection and tuning for the feature vectors produced by the document representation stage (Doc2Vec).

3.3. Doc2Vec with AutoML for Predatory Journal Classification

The predatory journal classification approach is illustrated in Figure 2. This study adopts the technique proposed by Feng et al. [20], which transforms the DOM tree into a textual corpus, followed by Doc2Vec representation learning and classification.

First, we collected the raw HTML pages of both legitimate and predatory journals and generated a corresponding corpus. The method begins by traversing the DOM tree of each HTML document to produce corpus text, enabling the Doc2Vec model to learn meaningful representations of HTML tag structures. Two traversal strategies were evaluated: Breadth-First Search (BFS) and Depth-First Search (DFS). The Doc2Vec model then learns from the transformed texts to capture informative corpus representations.

Subsequently, the trained Doc2Vec model is used to obtain vector representations, which serve as input for AutoML training. The AutoML framework automatically selects the most suitable classification model to achieve optimal performance in distinguishing between legitimate and predatory journals.

3.4. Dataset

The dataset was compiled by collecting links from both predatory and non-predatory journal websites. For predatory journals, the primary source was the Predatory Research Journals Data from Kaggle [18], which contains a total of 2,210 entries. In contrast, links to non-predatory journals were obtained from the Directory of Open Access Journals (DOAJ), comprising 21,269 entries.

Afterwards, we performed data preprocessing aimed to transform the HTML structure of journal websites into a textual format known as the DOM corpus, which is a sequential list of HTML tags derived from the <body> section of each webpage. This corpus serves as the main input for vectorization and model training. The entire process involved several interdependent stages: dataset integration, duplicate removal through URL normalization, web scraping, and DOM traversal using Depth-First Search (DFS) and Breadth-First Search (BFS) techniques.

Datasets of predatory and non-predatory journals shown by Table 1 were obtained from two different sources. These were merged into a unified dataset to ensure consistency in format and facilitate downstream analysis. The resulting dataset contained 23,479 entries, including 2,210 predatory and 21,269 non-predatory journals. Table 1 illustrates samples of this merged dataset in a structured format.

To ensure data cleanliness, duplicate entries were removed based on journal URLs. First, URL normalization was applied by converting all characters to lowercase and removing trailing slashes, enabling accurate comparison across entries. After normalization, 780 duplicates were identified and eliminated. The resulting dataset comprised 22,699 unique entries, with 1,919 predatory and 20,780 non-predatory journals.

The next phase involved extracting the main HTML content from each journal website. Specifically, elements within the <body> tag were targeted, as they represent the core structural components of a webpage. Web scraping was automated by sending HTTP requests to each journal URL and parsing the returned content.

The extracted HTML content was stored in JSON format for further processing. In cases of failure due to issues such as invalid SSL certificates, HTTP errors (e.g., 404 Not Found, 403 Forbidden), DNS resolution failures, or connection timeouts the URLs were logged separately. Out of 22,699 entries, a total of 16,170 pages were successfully scrapped, including 1,350 predatory and 14,820 non-predatory journals. The remaining 6,529 entries failed due to the aforementioned issues.

The final stage involved transforming each scraped HTML page into its DOM representation. The structure of the DOM tree was traversed using two standard algorithms: Depth-First Search (DFS) and Breadth-First Search (BFS). Each traversal produced an ordered sequence of HTML tags (e.g., , <p>, <a>) from the <body> section, effectively converting each webpage into a textual sequence. These sequences formed the DOM corpus, which was later used for feature extraction and classification model training. By employing both DFS and BFS methods, the study enabled comparative analysis of traversal-based structural representations and their impact on model performance.

3.5. Exprimental Settings

The experiment setup included configuration of hardware, software, Doc2Vec parameters (vector size, epochs, etc.), and data split (80% training, 20% testing). For training the word vectors using Doc2Vec, a vector size of 100, a window size of 5, and 20 epochs were employed. An epoch value of 20 was selected as a balanced setting to allow sufficient learning of structural patterns in the DOM-tree traversal corpus while maintaining computational efficiency. Auto-Sklearn was used to automate model selection and hyperparameter tuning. To address class imbalance in the training set, SMOTENN was employed by combining SMOTE for oversampling with Edited Nearest Neighbours (ENN) for undersampling.

The complete source code used in this study, including data preprocessing, DOM traversal, vectorization, and model training, is publicly available at the following GitHub repository: https://github.com/ersankarimi/predatory-journal-classifier.

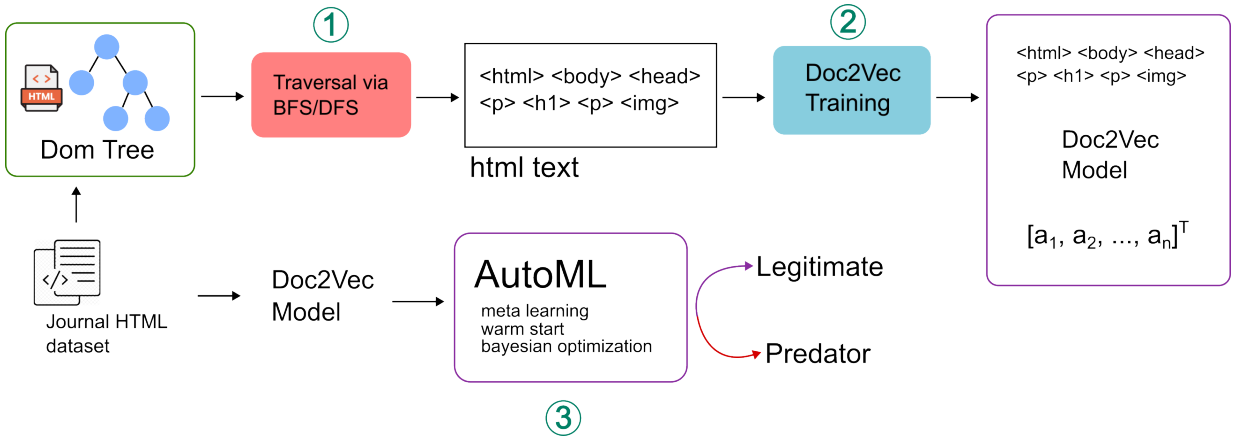

Figure 2: The proposed method, inspired by Feng et al. [20], for predatory journal detection, consisting of three main stages: (1) DOM tree traversal using breadth-first or depth-first search; (2) HTML tag corpus training; and (3) AutoML training using a Doc2Vec model.

Table 1: Example of Merged Dataset of Predatory and Non-Predatory Journals

journal_title	journal_url	is_predatory
Prolíngua	http://periodicos.ufpb.br/index.php/prolingua/index	0
World Scientific News	http://www.worldscientificnews.com/	1
Yangtze Medicine	https://www.scirp.org/journal/ym	1

4. Results and Discussions

This study evaluates the impact of DOM traversal methods which are Depth-First Search (DFS) and Breadth-First Search (BFS) on the performance of a predatory journal classification model using a combination of Doc2Vec and AutoML. The evaluation focused on two metrics: accuracy and macro-average F1-score.

Web pages from predatory and non-predatory journal sources were collected, preprocessed, and converted into DOM corpora using DFS and BFS strategies. These corpora were then vectorized with Doc2Vec and classified via AutoSklearn. The training was conducted across varying time durations (15–90 minutes) to observe model performance dynamics.

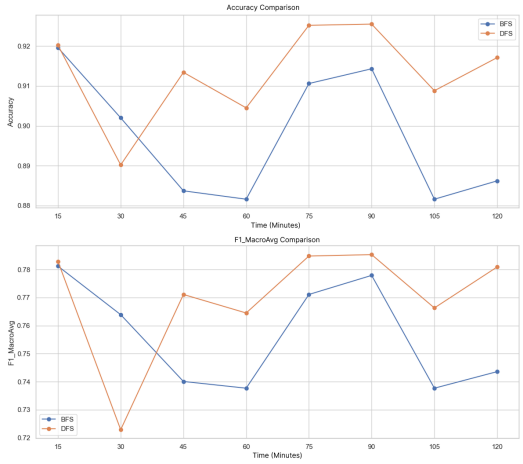

Figure 3: Accuracy and Macro F1-score Across Training Durations

The experimental results (see Figure 3) reveal that DFS generally outperforms BFS in both accuracy and macro F1-score across most training durations. The DFS model achieved its best performance at 90 minutes with an accuracy of 0.9255 and a macro F1-score of 0.7853. In contrast, BFS peaked earlier at 15 minutes with a slightly lower F1-score of 0.7812 and accuracy of 0.9196, but it displayed more fluctuation across training times.

Table 2: Precision and Recall Comparison between BFS and DFS Models

Class	Matrix	BFS (15 Minutes)	DFS (90 Minutes)
Predator	Precision	0.5128	**0.5413**
	Recall	**0.7444**	0.7037
Non-Predator	Precision	**0.9757**	0.9723
	Recall	0.9356	**0.9457**
Macro	Precision	0.7442	**0.7568**
	Recall	**0.8400**	0.8247

Table 2 shows a detailed precision-recall comparison. While BFS achieved a higher macro recall (0.8400), DFS attained superior macro precision (0.7568), indicating more accurate positive predictions. These results suggest that DFS benefits more from longer training durations and yields more stable performance, while BFS is faster but less consistent.

Although the difference is not statistically significant, the results highlight the potential of DFS in forming structurally meaningful representations for complex web content. Both traversal methods are viable, but DFS may be preferable when training resources and time allow. DFS provides higher performance than BFS, potentially because its traversal strategy better preserves local semantic relationships among neighboring DOM elements. For example, a tag such as `<h2>` is more likely to remain closely associated with related tags like `<p>` and `<b>` within the same subtree during corpus generation.

To further validate the effectiveness of the proposed Doc2Vec and AutoML pipeline, we compared it with a clustering-based baseline using Hierarchical Clustering followed by a Nearest Class Mean (NCM) classifier. This baseline was inspired by previous work on DOM-based web page classification, where DOM-derived document representations

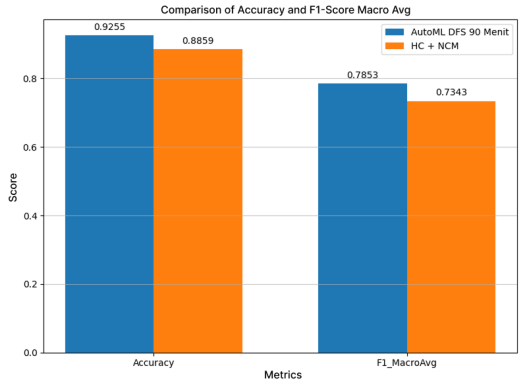

Figure 4: Overall performance comparison between Doc2Vec–AutoML (DFS, 90 minutes) and the HC–NCM baseline.

are first grouped via hierarchical clustering and classification is performed by assigning samples to the nearest class centroid (Nearest Class Mean) [20].

Figure 4 summarizes the overall performance of both approaches. The best Doc2Vec and AutoML model (DFS, 90 minutes) achieved an accuracy of 0.9255 and a macro avg F1-score of 0.7853. The discrepancy between accuracy and F1-score may be attributed to class imbalance in the testing set. While accuracy is influenced by the dominant class, the F1-score reflects classification performance on minority-class samples based on the computation of the harmonic mean of precision and recall.

In contrast, the baseline HC–NCM [20] obtained an accuracy of 0.8859 and a macro avg F1-score of 0.7343. This corresponds to an improvement of 0.0396 in accuracy and 0.0510 in macro F1-score, indicating that the AutoML-based approach provides more robust classification performance under the same DOM-derived representation.

5. Conclusion

The experimental results indicate that AutoML with Doc2Vec can be a baseline for automatic predatory journal classification based on the DOM structure. In constructing the similar text-based data for Doc2Vec, both DFS and BFS traversal can be utilized which influence the model's performance. The best DFS-based model was achieved with 90 minutes of training, yielding an accuracy of 0.9255 and a macro average F1-score of 0.7853. In contrast, the best BFS-based model was obtained within 15 minutes, with an accuracy of 0.9196 and a macro F1-score of 0.7812.

The limitation of this study is that the model only accounts for the HTML of the homepage to detect predatory journals and uses a two-stage approach consisting of corpus training for HTML tag representation and AutoML for prediction. Future studies should investigate beyond the homepage and incorporate multi-modal data, as several pages may improve the representation for the classification of predatory journals. In addition, utilizing images and other web-based data could potentially improve the performance.

6. Acknowledgments

Acknowledgements section if necessary

References

[1] N. M. Harum Harahap, "Tren saat ini dan masalah dalam akses open akses dan komunikasi ilmiah," *IQRA': Jurnal Ilmu Perpustakaan dan Informasi (e-Journal)*, vol. 14, no. 1, p. 63, Feb. 2020.

[2] H. Heriyanto, "Research into open access: Impact and user perspective," *Anuva*, vol. 3, no. 2, pp. 95–100, Jun. 2019.

[3] T. M. Mahmud, "INFOMASI ILMIAH OPEN ACCESS_bentuk DAN PENGARUHNYA UNTUK CIVITAS AKADEMIK," *BIBLIOTIKA : Jurnal Kajian Perpustakaan dan Informasi*, vol. 4, no. 1, pp. 10–17, Jul. 2020. [Online]. Available: https://journal2.um.ac.id/index.php/bibliotika/article/view/14752

[4] "Predatory and Questionable Publishing Practices," Apr. 2024. [Online]. Available: https://library.maastrichtuniversity.nl/research/publishing/information/publishing-strategy/predatory-and-questionable-publishing-practices/

[5] J. Fagan-Fry, "NOAA Library: Journal Evaluation & Predatory Publishing: Home." [Online]. Available: https://library.noaa.gov/predatorypublishing/home

[6] M.-C. Roland, "Publish and perish," *The EMBO Reports*, vol. 8, no. 5, pp. 424–428, 2007.

[7] D. Butler, "Investigating journals: The dark side of publishing," *Nature*, vol. 495, no. 7442, pp. 433–435, 2013.

[8] S. Eriksson and G. Helgesson, "The false academy: predatory publishing in science and bioethics," *Medicine, Health Care and Philosophy*, vol. 20, no. 2, pp. 163–170, 2017.

[9] V. Macháček and M. Srholec, "Retraction note to: Predatory publishing in scopus: evidence on cross-country differences," *Scientometrics*, vol. 127, no. 3, pp. 1667–1667, Mar. 2022.

[10] J. Beall, "Predatory journals: Ban predators from the scientific record," *Nature*, vol. 534, no. 7607, p. 326, Jun. 2016.

[11] M. Feurer, A. Klein, K. Eggensperger, J. T. Springenberg, M. Blum, and F. Hutter, *Auto-sklearn: Efficient and Robust Automated Machine Learning*. Cham: Springer International Publishing, 2019, pp. 113–134. [Online]. Available: https://doi.org/10.1007/978-3-030-05318-5_6

[12] J. Bohannon, "Who's Afraid of Peer Review?" *Science*, vol. 342, no. 6154, pp. 60–65, Oct. 2013. [Online]. Available: https://www.science.org/doi/10.1126/science.342.6154.60

[13] J. Beall, "Predatory publishers are corrupting open access," *Nature*, vol. 489, no. 7415, p. 179, Sep. 2012.

[14] L.-X. Chen, K.-S. Wong, C.-H. Liao, and S.-M. Yuan, "Predatory journal classification using machine learning," in *2020 3rd IEEE International Conference on Knowledge Innovation and Invention (ICKII)*, 2020, pp. 193–196.

[15] A. Adnan, S. Anwar, T. Zia, S. Razzaq, F. Maqbool, and M. Z. U. Rehman, " Beyond Beall's Blacklist: Automatic Detection of Open Access Predatory Research Journals ," in *2018 IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*. Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2018, pp. 1692–1697. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/HPCC/SmartCity/DSS.2018.00274

[16] W. M. B. Ateeq and H. S. Al-Khalifa, "Intelligent framework for detecting predatory publishing venues," *IEEE Access*, vol. 11, pp. 20 582–20 618, 2023.

[17] G. Sharma, V. Tripathi, and V. Singh, "A novel ai system for detecting academic predatory practices using big data," *Multidisciplinary Science Journal*, vol. 7, no. 11, pp. 2 025 519–2 025 519, 2025.

[18] J. A. Teixeira da Silva and G. Kendall, "(mis-)classification of 17,721 journals by an artificial intelligence predatory journal detector," *Publishing Research Quarterly*, vol. 39, no. 3, pp. 263–279, Sep 2023. [Online]. Available: https://doi.org/10.1007/s12109-023-09956-y

[19] J. Wu, T. Liu, K. Mu, and L. Zhou, "Identification and causal analysis of predatory open access journals based on interpretable machine learning," *Scientometrics*, vol. 129, no. 4, pp. 2131–2158, Apr 2024. [Online]. Available: https://doi.org/10.1007/s11192-024-04969-6

[20] J. Feng, Y. Zhang, and Y. Qiao, "A detection method for phishing web page using DOM-based Doc2Vec model," *J. Comput. Inf. Technol.*, vol. 28, no. 1, pp. 19–31, Jul. 2020.

[21] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 1188–1196. [Online]. Available: https://proceedings.mlr.press/v32/le14.html

[22] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," *Advances in neural information processing systems*, vol. 28, 2015.

[23] M. Feurer, K. Eggensperger, S. Falkner, M. Lindauer, and F. Hutter, "Auto-sklearn 2.0: Hands-free automl via meta-learning," *Journal of Machine Learning Research*, vol. 23, no. 261, pp. 1–61, 2022.

