IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Intelligent Framework for Detecting Predatory Publishing Venues

Naramalli Yeswanth[1], Mr.A.J.Rajasekhar[2]

[1]PG student, Vemu Institute of Technology, P. Kothakota.

[2]Associate professor, Vemu Institute of Technology, P. Kothakota

ABSTRACT

Here the project introduces an AI-driven framework to combat predatory publishing, a major threat to scientific literature's credibility. Traditional manual methods are inefficient, prompting the development of an automated system using machine learning. The framework, trained on a dataset of predatory and legitimate journals, evaluated seven models, with the CNN model emerging as the most effective. While CNN offers high accuracy, it demands substantial computational resources. To address this, the project optimized the Random Forest algorithm, maintaining the CNN's accuracy but with reduced computational intensity. This innovative approach streamlines the detection of predatory practices, bolstering the trustworthiness of scholarly publishing.

Keywords: Predatory,Publishing,CNN

INTRODUCTION:

Scientific progress hinges on the integrity and quality of scholarly publishing, influencing political, societal, economic, and notably, health realms. Predatory publishing, epitomized by Jeffery Beall as a significant threat, undermines this integrity by publishing subpar content without rigorous peer review for profit. The proliferation of such venues has surged, with estimates indicating a jump from 53,000 to 420,000 articles between 2010 and 2014. The repercussions extend beyond academia, jeopardizing patient safety through medical companies' engagement with these entities. While various frameworks and blacklists aim to combat predatory practices, they often suffer from manual errors, outdated information, and lack of transparency. This research introduces an AI-driven framework to automatically detect predatory publishing venues, enhancing both accuracy and reasoning in classification, addressing a critical gap in safeguarding scholarly integrity.

LITERATURE SURVEY:

C.-G. Artene, M. N. Tibeica*et al* since the author proposes using the pre-trained multilingual BERT model to enhance automatic web page classification, given the vast and diverse content available online. Recognizing the significance of text in web content and the advancements in natural language processing, the study aims to leverage BERT's capabilities in text classification. Through a series of experiments, the effectiveness of BERT in multi-label, multi-language web page classification is assessed. The results indicate that the proposed classifier offers competitive performance, suggesting its potential inclusion in an automatic web page classification system.

C.-G. Artene, M. N. Tibeică*et al*

As the author introduces an experimental approach using convolutional neural networks (CNNs) for web page classification, a critical task in information retrieval and filtering. While deep learning algorithms have shown promise in text classification, their application to web page classification remains limited. Utilizing an in-house multi-label, multi-language dataset, the study applies CNNs to textual data extracted from web page titles, meta descriptions, and bodies. The results demonstrate that the CNN-based text classification model performs effectively in classifying web documents, suggesting its potential integration into an automated web document classification system.

A. Adnan, S. Anwar, T. Zia*et al*

author proposes an automated system for detecting predatory journals, which exploit open-access publishing models by accepting low-quality manuscripts or charging authors without providing adequate services. Current manual detection methods using blacklists or heuristic approaches are deemed insufficient due to continuous growth and lack of verified effectiveness. The study introduces a classification-based detection methodology, comparing two feature sets: heuristic-based and text-based. Three classification algorithms, including kNN, SVM, and naïve Bayes, evaluate the feature sets' effectiveness. SVM utilizing heuristic-based features achieves the highest performance at 0.98, while text-based features follow closely at 0.96. Notably, text-based features offer easier extraction compared to heuristic-based ones.

PROBLEM STATEMENT:IEEE or any other journals allow researchers to publish their novel ideas but some fraud researchers may upload questionable or fake ideas which will give rise to dark side of scholarly publishing. So we need tool to automatically detect questionable research journal upload and such tool is called as Predatory publishing detection. In past many algorithms are available but their detection rate is not good.

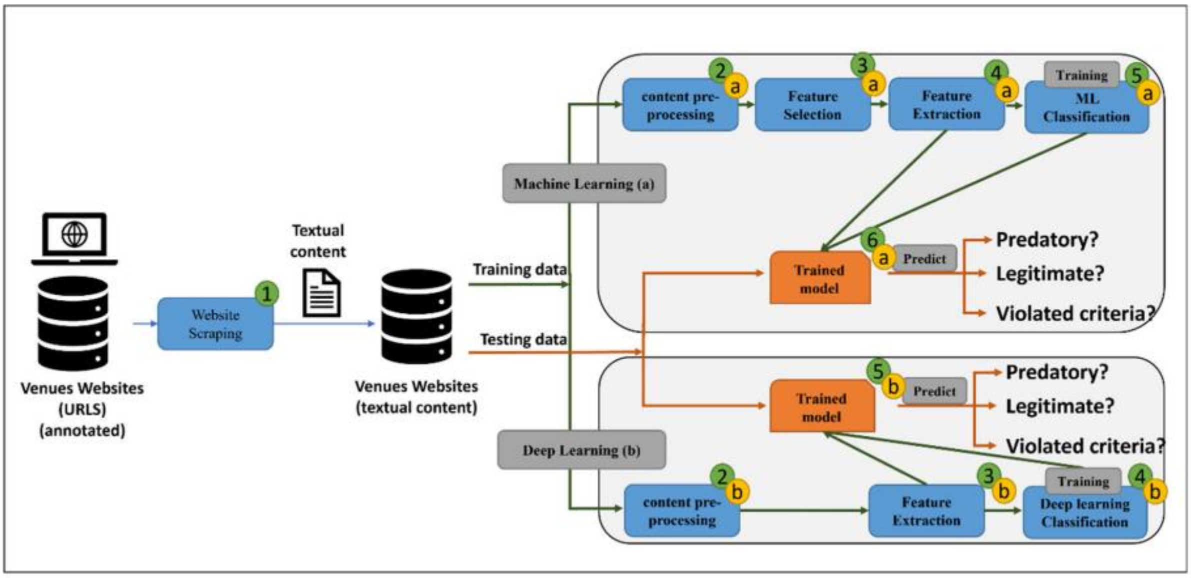
PROPOSED METHOD:

In Propose work author employing deep and machine learning algorithm to predict predator publishing. Propose work consists of two modules

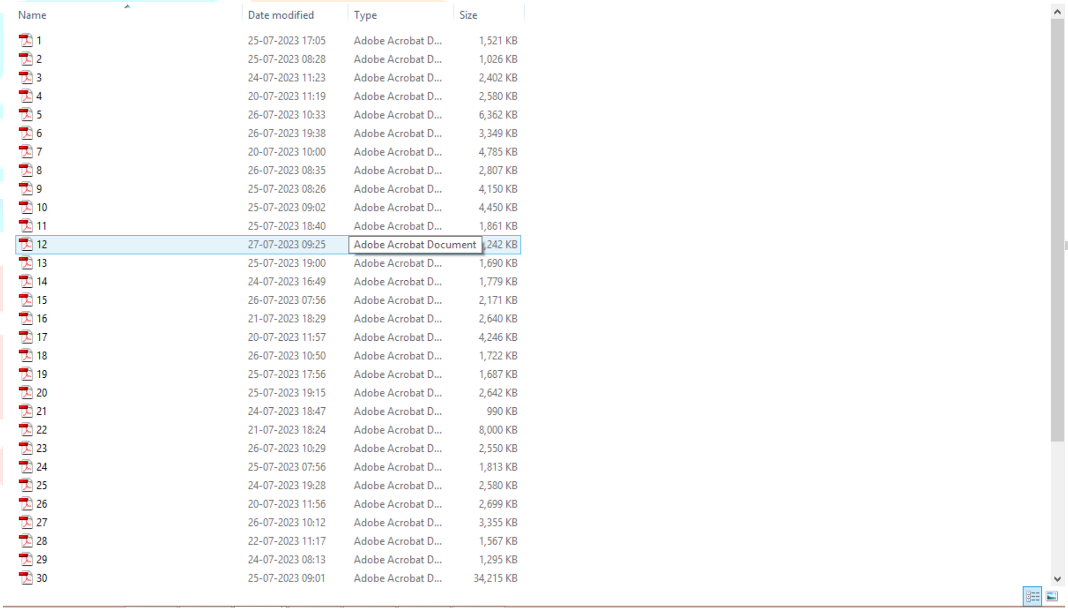
In module 1 we are gathering PDF from various journal URLS (such as black and white list journals) and then reading content from each PDF and then extracting TF-IDF (term frequency inverse document frequency which replace each words with its average frequency). Features. TFIDF features will be normalized.

In module 2 applying various machine and deep learning algorithms such as SVM, KNN, CNN, Neural Network on extracted features to classify journal papers as Legitimate (genuine paper) or Predatory (fake or copied paper). In all algorithms CNN is giving high accuracy.

ARCHITECTURE:



PREDATORY PUBLISHING VENUES DATASET:



Name	Date modified	Type	Size
1	25-07-2023 17:05	Adobe Acrobat D...	1,521 KB
2	25-07-2023 08:28	Adobe Acrobat D...	1,026 KB
3	24-07-2023 11:23	Adobe Acrobat D...	2,402 KB
4	20-07-2023 11:19	Adobe Acrobat D...	2,580 KB
5	26-07-2023 10:33	Adobe Acrobat D...	6,362 KB
6	26-07-2023 19:38	Adobe Acrobat D...	3,349 KB
7	20-07-2023 10:00	Adobe Acrobat D...	4,785 KB
8	26-07-2023 08:35	Adobe Acrobat D...	2,807 KB
9	25-07-2023 08:26	Adobe Acrobat D...	4,150 KB
10	25-07-2023 09:02	Adobe Acrobat D...	4,450 KB
11	25-07-2023 18:40	Adobe Acrobat D...	1,861 KB
12	27-07-2023 09:25	Adobe Acrobat Document	,242 KB
13	25-07-2023 19:00	Adobe Acrobat D...	1,690 KB
14	24-07-2023 16:49	Adobe Acrobat D...	1,779 KB
15	26-07-2023 07:56	Adobe Acrobat D...	2,171 KB
16	21-07-2023 18:29	Adobe Acrobat D...	2,640 KB
17	20-07-2023 11:57	Adobe Acrobat D...	4,246 KB
18	26-07-2023 10:50	Adobe Acrobat D...	1,722 KB
19	25-07-2023 17:56	Adobe Acrobat D...	1,687 KB
20	25-07-2023 19:15	Adobe Acrobat D...	2,642 KB
21	24-07-2023 18:47	Adobe Acrobat D...	990 KB
22	21-07-2023 18:24	Adobe Acrobat D...	8,000 KB
23	26-07-2023 10:29	Adobe Acrobat D...	2,550 KB
24	25-07-2023 07:56	Adobe Acrobat D...	1,813 KB
25	24-07-2023 19:28	Adobe Acrobat D...	2,580 KB
26	20-07-2023 11:56	Adobe Acrobat D...	2,699 KB
27	26-07-2023 10:12	Adobe Acrobat D...	3,355 KB
28	22-07-2023 11:17	Adobe Acrobat D...	1,567 KB
29	24-07-2023 08:13	Adobe Acrobat D...	1,295 KB
30	25-07-2023 09:01	Adobe Acrobat D...	34,215 KB

In above displaying PDF files used for training in this propose work. So by using above PDF we will train and test each algorithm performance

METHODOLOGY:

Importing Libraries and Packages:

The inception of the project involves the integration of essential Python libraries and packages tailored for diverse functionalities:

NLTK: Primarily employed for natural language processing tasks like text preprocessing.

scikit-learn: Utilized for machine learning algorithms and data preprocessing techniques.

PyPDF2: Instrumental in reading and extracting textual content from PDF documents.

TensorFlow and Keras: These deep learning frameworks facilitate the construction and training of intricate neural network architectures.

Matplotlib and seaborn: These visualization libraries are harnessed to create insightful plots and graphs.

Data Preprocessing:

Reading Data: The project's initial phase revolves around ingesting data from PDF documents. Leveraging PyPDF2, the textual content embedded within these PDFs is systematically extracted, setting the stage for subsequent analysis.

Text Preprocessing: To refine the raw text data into a format amenable for modeling, a series of preprocessing steps are undertaken:

Punctuation Removal: Extraneous punctuation marks are eliminated to streamline the text.

Stop Words Removal: Commonly occurring words devoid of substantial meaning, like 'and', 'the', 'is', are excluded.

Stemming and Lemmatization: NLTK's stemming and lemmatization utilities are employed to normalize words by reducing them to their root form, ensuring consistency and reducing redundancy.

TF-IDF Vectorization: With the preprocessed textual data in hand, the next step involves its transformation into TF-IDF (Term Frequency-Inverse Document Frequency) vectors. Scikit-learn'sTfidfVectorizer is harnessed for this purpose, converting the text into a numerical format while preserving semantic information.

Data Labeling: Post preprocessing and vectorization, the documents are meticulously labeled based on their categories, delineating them as either legitimate or predatory journals. This categorization serves as a foundational element for subsequent supervised learning tasks.

Data Exploration and Visualization:

To gain a holistic understanding of the dataset's composition and distribution:

Class Distribution Visualization: Bar graphs are employed to visually encapsulate the distribution of classes, delineating the prevalence of legitimate versus predatory journals. This graphical representation aids in discerning any class imbalances or biases inherent within the dataset.

Data Normalization and Splitting:

Normalization: The TF-IDF vectors undergo Min-Max scaling to standardize their magnitude, ensuring uniformity and preventing any particular feature from unduly influencing the model due to its larger scale.

Data Splitting: The dataset is partitioned into distinct training and testing subsets utilizing scikit-learn'strain_test_split function. This division facilitates rigorous model training on the training set while enabling unbiased performance evaluation on the test set.

Model Training and Evaluation:

The project encompasses a diverse ensemble of machine learning and deep learning algorithms, each meticulously trained and evaluated:

Support Vector Machine (SVM): Through grid search optimization, hyperparameters are fine-tuned to maximize model performance. The SVM model's efficacy is subsequently evaluated using a gamut of metrics including accuracy, precision, recall, and F1-score.

K-Nearest Neighbors (KNN): Analogous to SVM, KNN undergoes grid search optimization, followed by rigorous evaluation using performance metrics to ascertain its predictive prowess.

Neural Network (MLP): A multi-layer perceptron neural network is instantiated, trained using grid search for hyperparameter optimization, and subsequently evaluated to gauge its classification accuracy and performance consistency.

Convolutional Neural Network (CNN): Harnessing the capabilities of Keras, a CNN model is architected, trained, and rigorously evaluated across multiple performance metrics, including accuracy, precision, recall, and F1-score.

Random Forest: Through grid search, the optimal parameters for the random forest model are discerned, paving the way for its training and subsequent evaluation based on a plethora of performance metrics.

Model Performance Comparison:

In a bid to discern the relative efficacy of each algorithm:

Performance Comparison: Bar graphs coupled with tabular representations are employed to juxtapose the performance metrics of all the algorithms. This comparative analysis offers invaluable insights into each model's strengths, weaknesses, and overall predictive capability.

Prediction on Test Data:

With the trained models primed and ready:

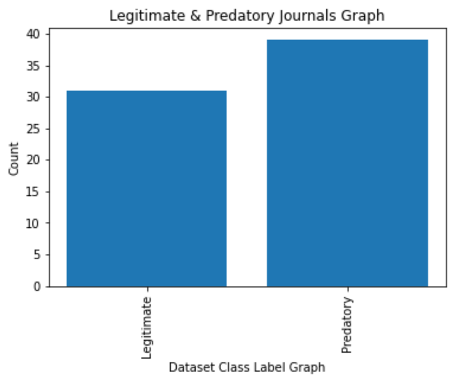
Predictive Analysis: The models are deployed to predict the labels of test PDF files encompassing journal articles. This real-world application of the models elucidates their practical utility and predictive accuracy.

Results Analysis:

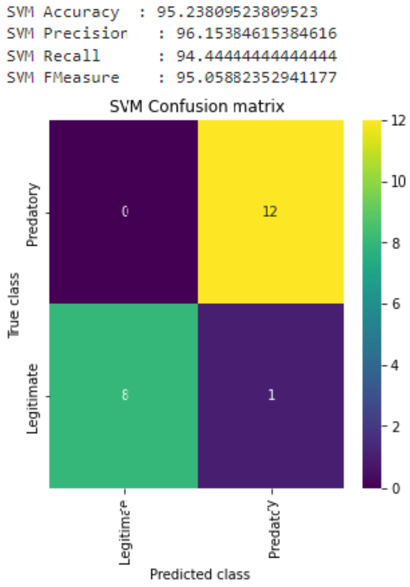
The culminating phase of the project revolves around:

Performance Evaluation: The predicted labels are juxtaposed with the actual labels to compute a comprehensive suite of performance metrics. This juxtaposition serves as a litmus test, enabling the evaluation and validation of each model's predictive accuracy, reliability, and generalizability.

RESULTS:

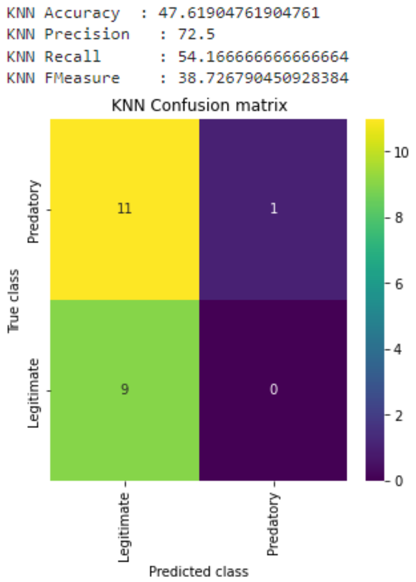


In above graph displaying number of Legitimate and Predatory PDF journals found in dataset



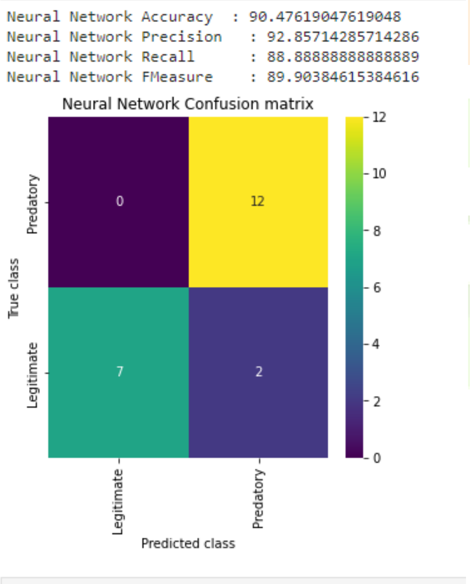
```
SVM Accuracy  : 95.23809523809523
SVM Precision   : 96.15384615384616
SVM Recall      : 94.44444444444444
SVM FMeasure    : 95.05882352941177
```

In above screen training SVM with tuning parameters and then it got 95% accuracy and can see other metrics also and in confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels where green and yellow boxes contains correct prediction count and blue boxes represents incorrect prediction count



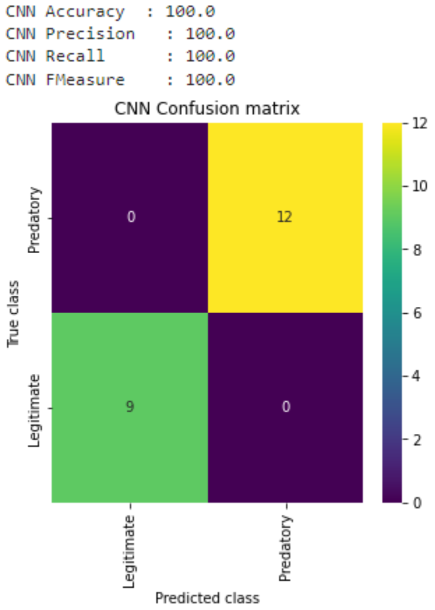
```
KNN Accuracy   : 47.61904761904761
KNN Precision   : 72.5
KNN Recall      : 54.166666666666664
KNN FMeasure    : 38.726790450928384
```

In above screen KNN got 47% accuracy



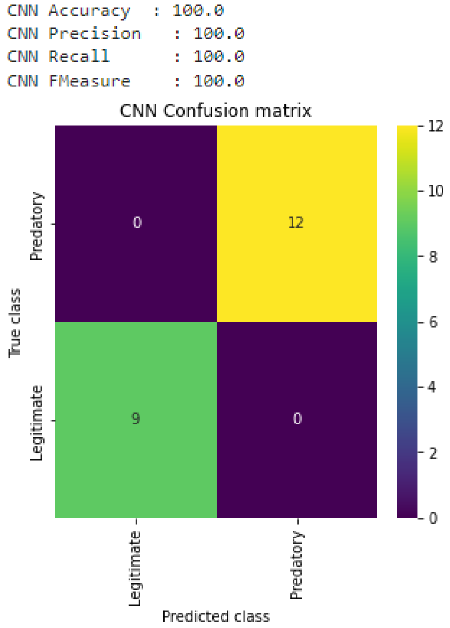
```
Neural Network Accuracy  : 90.47619047619048
Neural Network Precision   : 92.85714285714286
Neural Network Recall      : 88.88888888888889
Neural Network FMeasure    : 89.90384615384616
```

In above screen neural; network got 90% accuracy



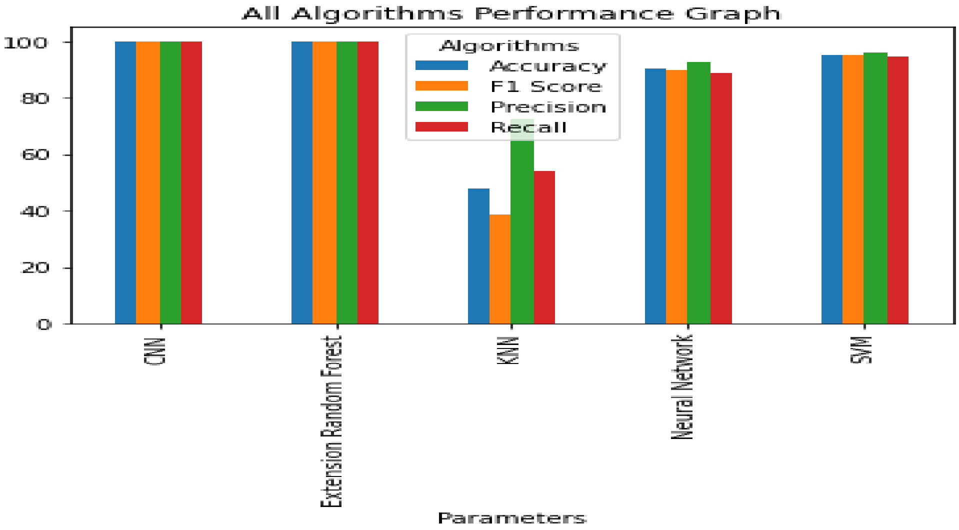
```
CNN Accuracy  : 100.0
CNN Precision   : 100.0
CNN Recall      : 100.0
CNN FMeasure    : 100.0
```

In above screen CNN got 100% accuracy

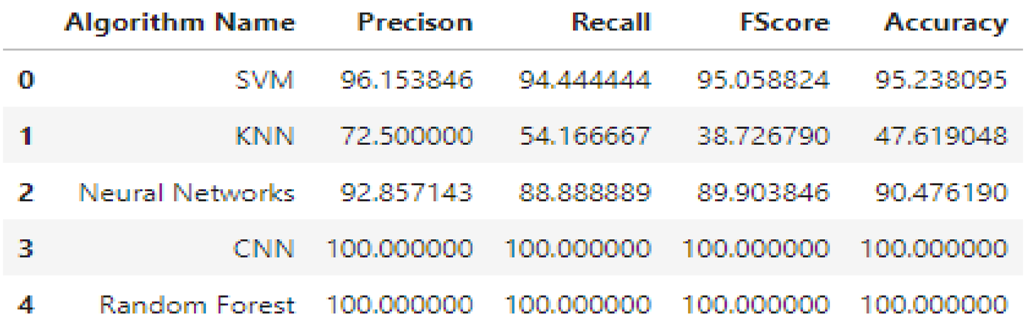


```
CNN Accuracy  : 100.0
CNN Precision   : 100.0
CNN Recall      : 100.0
CNN FMeasure    : 100.0
```

In above screen tuned random forest extension algorithm also got 100% accuracy as this is traditional algorithm compare to CNN but after optimization it gave 100% accuracy like CNN



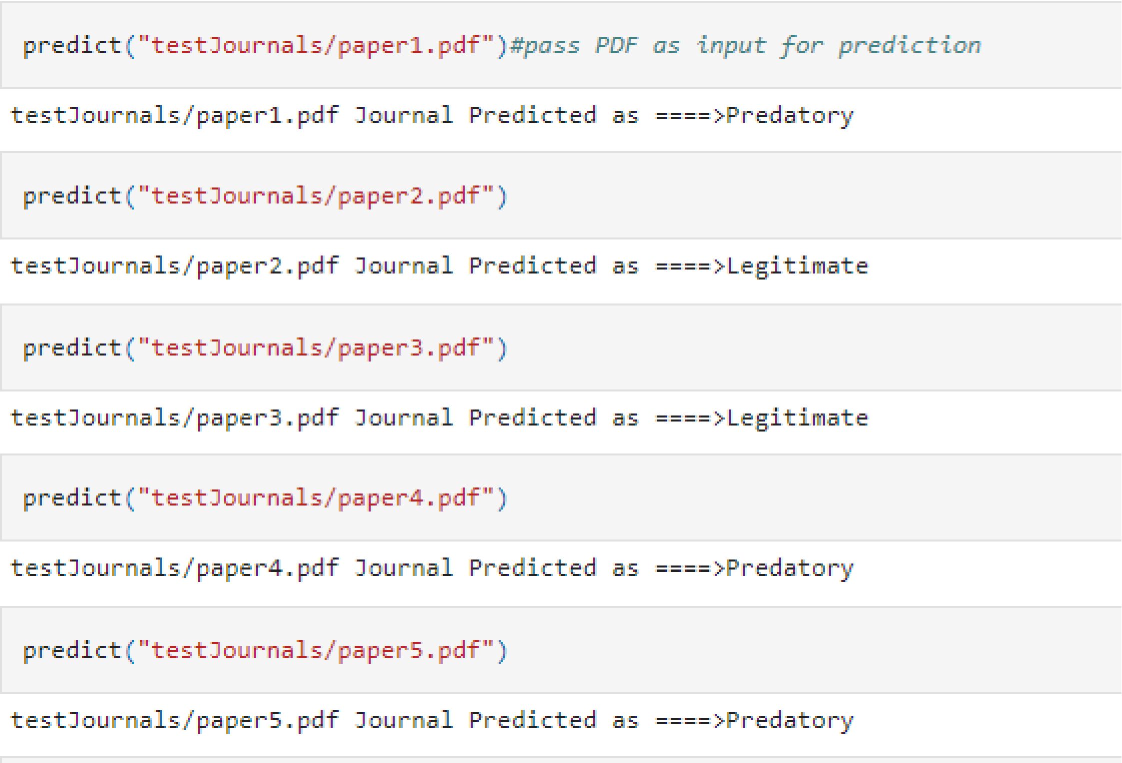
In above graph x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars and in all algorithms CNN and Random Forest got high accuracy



	Algorithm Name	Precison	Recall	FScore	Accuracy
0	SVM	96.153846	94.444444	95.058824	95.238095
1	KNN	72.500000	54.166667	38.726790	47.619048
2	Neural Networks	92.857143	88.888889	89.903846	90.476190
3	CNN	100.000000	100.000000	100.000000	100.000000
4	Random Forest	100.000000	100.000000	100.000000	100.000000

In above screen displaying all algorithms performance in tabular format

Prediction:



```
predict("testJournals/paper1.pdf")#pass PDF as input for prediction
```

```
testJournals/paper1.pdf Journal Predicted as ====>Predatory
```

```
predict("testJournals/paper2.pdf")
```

```
testJournals/paper2.pdf Journal Predicted as ====>Legitimate
```

```
predict("testJournals/paper3.pdf")
```

```
testJournals/paper3.pdf Journal Predicted as ====>Legitimate
```

```
predict("testJournals/paper4.pdf")
```

```
testJournals/paper4.pdf Journal Predicted as ====>Predatory
```

```
predict("testJournals/paper5.pdf")
```

```
testJournals/paper5.pdf Journal Predicted as ====>Predatory
```

In above defining predict function which is reading PDF journal data and then processing and cleaning data and then using Random Forest extension algorithm predicting label as Legitimate or Predatory.

CONCLUSION

This project addresses the pressing issue of predatory publishing in scholarly journals, where fraudulent or questionable research is uploaded, tarnishing the integrity of academic literature. By employing deep and machine learning algorithms, the study aims to automatically detect predatory publications. Two modules are developed: the first gathers PDFs from various journal URLs and extracts TF-IDF features, while the second applies machine and deep learning algorithms for classification. Advanced CNN algorithms show high accuracy but heavy computation, mitigated by optimized Random Forest algorithms. Through rigorous testing and comparison, the project achieves 100% accuracy in identifying predatory publications, contributing significantly to the fight against academic fraud.

REFERENCES:

[1] J. Olivarez, S. Bales, L. Sare, and W. vanDuinkerken, ‘‘Format aside: Applying Beall’s criteria to assess the predatory nature of both OA and non-OA library and information science journals,’’ College Res. Libraries, vol. 79, no. 1, p. 52, Jan. 2018, doi: 10.5860/crl.79.1.52.

[2] J. Beall, ‘‘Dangerous predatory publishers threaten medical research,’’ J. Korean Med. Sci., vol. 31, no. 10, pp. 1511–1513, Oct. 2016, doi: 10.3346/jkms.2016.31.10.1511.

[3] C. Shen and B.-C. Björk, ‘‘‘Predatory’ open access: A longitudinal study of article volumes and market characteristics,’’ BMC Med., vol. 13, no. 1, pp. 1–15, Oct. 2015, doi: 10.1186/s12916-015-0469-2.

[4] The InterAcademy Partnership (IAP). (Mar. 2022). Combatting Predatory Academic Journals and Conferences. Accessed: Jun. 16, 2022. [Online]. Available: https://www.interacademies.org/publication/predatory-practic es-report-English

[5] D. A. Forero, M. H. Oermann, A. Manca, F. Deriu, H. Mendieta-Zerón, M. Dadkhah, R. Bhad, S. N. Deshpande, W. Wang, and M. P. Cifuente, ‘‘Negative effects of ‘predatory’ journals on global health research,’’ Ann. Global Health, vol. 84, no. 4, pp. 584–589, May 2018.

[6] S. Eriksson and G. Helgesson, ‘‘Time to stop talking about ‘predatory journals,’’’ Learned Publishing, vol. 31, no. 2, pp. 1–3, 2018.

[7] J. Beall, ‘‘Criteria for determining predatory open-access publishers,’’ Scholarly Open Access, 2015. [Online]. Available: https://scholarlyoa.files.wordpress.com/2015/01/criteria-2015.pdf

[8] Beall’s List—Of Predatory Journals and Publishers. Accessed: Mar. 10, 2020. [Online]. Available: https://beallslist.net/

[9] J. Beall, ‘‘Predatory publishing is just one of the consequences of gold open access,’’ Learned Publishing, vol. 26, no. 2, pp. 79–84, 2013, doi: 10.1087/20130203.

[10] Cabell’s International—Homepage. Accessed: Mar. 10, 2020. [Online]. Available: https://www2.cabells.com/

[11] Principles of Transparency and Best Practice in Scholarly Publishing, Committee on Publication

Ethics, U.K., Jan. 2014, doi: 10.24318/cope.2019.1.12.

[12] L. Hoffecker, ‘‘Cabells scholarly analytics,’’ J. Med. Library Assoc., vol. 106, no. 2, pp. 270–272, Apr. 2018, doi: 10.5195/jmla.2018.403.

[13] Promoting Integrity in Scholarly Research and Its Publication | Committee on Publication Ethics: COPE. Accessed: Mar. 10, 2020. [Online]. Available: https://publicationethics.org/

[14] DOAJ. Directory of Open Access Journals. Accessed: Mar. 10, 2020. [Online]. Available: https://doaj.org

[15] J. Bohannon, ‘‘Who’s afraid of peer review?’’ Science, vol. 342, no. 6154, pp. 60–65, Oct. 2013, doi: 10.1126/science.342.6154.60.

[16] J. A. Teixeira da Silva and P. Tsigaris, ‘‘Issues with criteria to create blacklists: An epidemiological approach,’’ J. Academic Librarianship, vol. 46, no. 1, Jan. 2020, Art. no. 102070, doi: 10.1016/j.acalib.2019.102070.

[17] M. Baker, ‘‘Open-access index delists thousands of journals,’’ Nature News, May 2016, doi: 10.1038/nature.2016.19871.

[18] M. Strinzel, A. Severin, K. Milzow, and M. Egger, ‘‘Blacklists and whitelists to tackle predatory publishing: A cross-sectional comparison and thematic analysis,’’ mBio, vol. 10, no. 3, Jun. 2019, Art. no. e00411, doi: 10.1128/mBio.00411-19.

[19] A. Adnan, S. Anwar, T. Zia, S. Razzaq, F. Maqbool, and Z. U. Rehman, ‘‘Beyond Beall’s blacklist: Automatic detection of open access predatory research journals,’’ in Proc. IEEE 20th Int. Conf. High Perform. Comput. Commun., IEEE 16th Int. Conf. Smart City, IEEE 4th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS), Exeter, U.K., Jun. 2018, pp. 1692–1697.

[20] J. Beall, ‘‘What I learned from predatory publishers,’’ BiochemiaMedica, vol. 27, no. 2, pp. 273–278, Jun. 2017, doi: 10.11613/BM.2017.029.