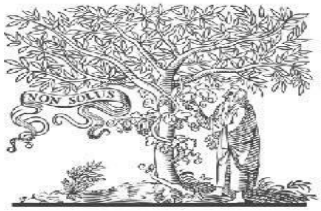
International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

COPY RIGHT



ELSEVIER SSRN

IJIEMR Transactions, online available on 31st Mar 2023. Link :http://www.ijiemr.org/downloads.php?vol=Volume-12&issue=Issue 03

10.48047/IJIEMR/V12/ISSUE 03/76

Title **PLAGIARISM DETECTION IN SOURCE CODES**

Volume 12, ISSUE 03, Pages: 535-541

Paper Authors
Mr.Naga Satish Kumar.M, Bhanu Naga Datha Sai.K, Devi Satya Vaishnavi.J, Sirisha.K

USE THIS BARCODE TO ACCESS YOUR ONLINE PAPER

To Secure Your Paper As Per UGC Guidelines We Are Providing A Electronic Bar Code

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

PLAGIARISM DETECTION IN SOURCE CODES

Mr.Naga Satish Kumar.M[1], Assistant Professor, Computer Science And Engineering
Seshadri Rao Gudlavalleru Engineering College
Gudivada, India.
Bhanu Naga Datha Sai.K[2], Student, Computer Science And Engineering
Seshadri Rao Gudlavalleru Engineering College
Gudivada, India.
Devi Satya Vaishnavi.J[3], Student, Computer Science And Engineering
Seshadri Rao Gudlavalleru Engineering College
Gudivada, India.
Sirisha.K[4], Student, Computer Science And Engineering
Seshadri Rao Gudlavalleru Engineering College
Gudivada, India.
Sri Hari.J[5], Student, Computer Science And Engineering
Seshadri Rao Gudlavalleru Engineering College
Gudivada, India.

Abstract
Recently, there have been several conversations about plagiarism in the technology and education sectors. Due to their extensive use and accessibility, electronic materials are easy for students, authors, and even academics to employ.To obtain knowledge without giving proper credit, use it, or incorporate it into one's own work. The problem, which endangers the educational process, is getting worse and worse.

There are several tools that deal with the problem of automating plagiarism detection, each of which has benefits and drawbacks of its own. Although it has the one downside of taking a lot of time, the conventional approach of detecting plagiarism through free text search using search engines is considered to be accurate and cost-free. Developing classification is necessary. We employ a range of deep learning techniques for plagiarism detection in order to create classification models and regression trees, evaluate the performance of these models, and identify plagiarism.

Keywords: Deep learning, plagiarism Detection, Cosine Similarity

Introduction

Artificial intelligence (AI) as in type of deep learning allows software applications to forecast outcomes more accurately without having beenexplicitly programmed to do so. Machine learning algorithms forecast new output values using historical data as input.

The Latin term "plagiarius," which meaning hijacker, is where the word plagiarism comes from. The process of finding plagiarism in a text or a document is known as plagiarism detection. The method used to find plagiarism is determined by the type of plagiarism. The act of taking text, code, graphics, ideas, etc. from multiple sources and representing them as one's own without providing credit is referred to as plagiarism. Copying is a mildly common issue in coding classes. Since the 1970s, a few scholars have been absent from the CPD (code plagiarism detection) research. A duplicated code is one that has been created from another code with only a few small, frequent changes, according to Alan and James. Regular modifications typically involve text exchanges that do

not require a deep comprehension for the code.

Language, opinions, conclusions, written materials, visual design, computer-related programmes, drawings, charts, and graphics, as well as any other contemporary and innovative work created and presented by another person, are all examples of plagiarism in a work. It is possible to define plagiarism as one of the electronic offences, along with computerspam, hacking, computer viruses, phishing, and other crimes. Plagiarism can be defined as the act of copying entirely or partially another person's works without properly citing them. The removal of plagiarism from all educational disciplines requires the use of a programme that can detect plagiarism because it might be a difficult task in some educational fields. [1]

Literature Review

Plagiarism detection is an important area of research in the field of natural language processing and information retrieval. In recent years, with the increasing availability of digital text and the ease of copying and pasting, the problem of plagiarism has become more widespread. In response to this, researchers have developed various methods and techniques for detecting plagiarism in digital text.

One of the earliest approaches to plagiarism detection was based on textual similarity analysis. This approach involves comparing the text of a document against a set of reference documents to identify similarities or exact matches. This method has been widely used in plagiarism detection systems such as Turnitin, which uses a large database of academic papers to check for similarity in student submissions. [6]

Another approach to plagiarism detection is based on stylometric analysis, which involves analyzing the writing style of a document to identify patterns and characteristics that are unique to a particular author. This approach has been used in plagiarism detection systems such as JPlag, which uses a combination of stylometric and textual similarity analysis to detect plagiarism in programming code.

In recent years, machine learning approaches have gained popularity in plagiarism detection. These approaches involve training a machine learning model on a large dataset of documents to learn patterns of similarity and difference between documents. The model can then be used to detect plagiarism in new documents based on the patterns it haslearned. Some popular machine learning approaches to plagiarism detection include deep learning models such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs).

Despite the many advances in plagiarism detection research, there are still challenges and limitations to be addressed. For example, some plagiarism detection methods may be less effective at detecting paraphrasing or rephrasing of text rather than exact matches. Additionally, some methods may be more suitable for detecting plagiarism in certain types of text or domains, such as academic papers or programming code.

In conclusion, plagiarism detection is a challenging and important area of research in natural language processing and information retrieval. Textual similarity analysis, stylometric analysis, and machine learning approaches are all widely used in plagiarism detection systems, each with their own strengths and limitations. Future research in this area will continue to focus on developing more accurate and effective methods for detecting plagiarism in digital text.

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Existing System

The definition of source code plagiarism is a programme that has been duplicated with only a few commonplace changes. It is a serious problem in both academics and business. Instead of replicating writings, it is literally reusing programmes. A program's source code can be reused in a variety of ways, including by copying and pasting small sections of it or by copying big sections and using artistic masking to hide the original copied programme.

There are so many existing systems for the plagiarism detection.In olden days there is no plagiarism detection system so they only check all the codes written by the students or professionals and check whether the two codes are same or not and whether they copy the code from the online and it is plagiarized or not.In the existing system they uses the structure based methods and attribute counting techniques.

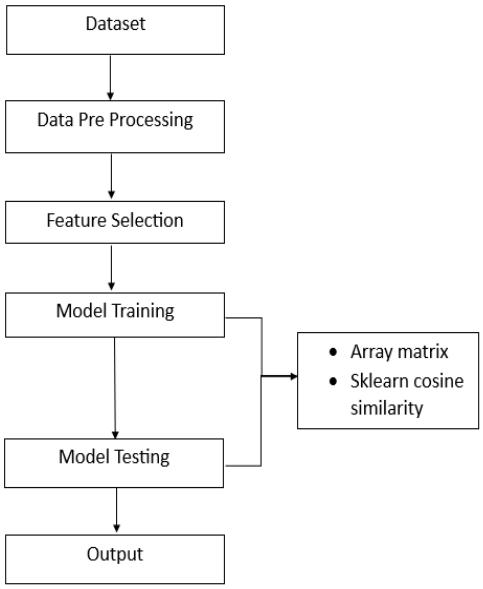
The structural similarity between two programmes can be used to determine their degree ofresemblance. When the level of plagiarism is four or above, it is challenging to identify it since the structured characteristics of copied manuscripts differ from the original document.so we cannot tell about plagiarism based on the structured methods.

In attribute counting technique by counting the number of operators and operands, the number of unique operators in a programme, the number of unique operands in a programme, the total number of operators in a programme, the total number of operands in a programme, the code lines (excluding comment lines) in a programme, the used and declared variables, and the total number of control statements in a programme, the attribute counting system relied on Halstead software features to identify plagiarism in student-generated programmes.It has low accuracy and the predections are somewhat false.It can't detect the plagiarism correctly.They also used the xplag technique but it also has the ow accuracy and it cant predicts the plagiarism correctly. [7]

Proposed System

To overcome the drawbacks in the existing system we proposed a new system to detect the plagiarism. In this we used a dataset which contain the codes and here we used some algorithms. They are array matrix and cosine similarity. The pre-trained model receives the input data which is the pre processed data .Here we take the source codes data from the Kaggle website and we applies the algorithms on it.In the dataset there are different codes that we have given.

On the data set we apply the array matrix and sklearn cosine similarity algorithm to detect the plagiarism. After the algorithm is applied on the data set we take an input from the user and based on the algorithms we detect whether the algorithm is plagiarised or not and we also detect the percentage of the plagiarism.Here we categorized the plagiarism into 3 types.They are low plagiarized,medium plagiarized and highly plagiarised based on the percentage of plagiarism in it.This proposed system helps us to detect the plagiarism very easily. [2]

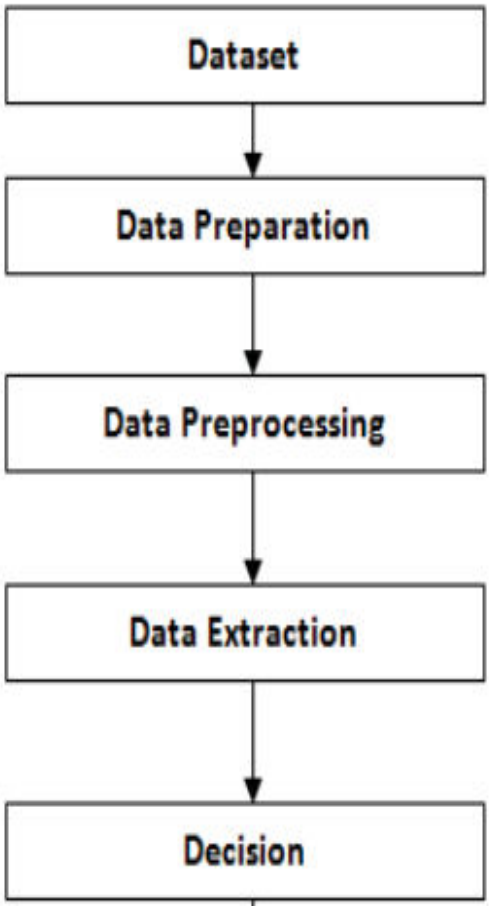


We offered many machine learning techniques in this study to know about plagiarism detection. In order to detect plagiarism in the source codes, a variety of models are used. These models include

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

array matrix and sklearn cosine similarity.

The following are our steps or algorithm steps:



A.Array Matrix

In machine learning, arrays are the most significant family of higher-order collections. Images, text files, and many other types of data are represented by them.An array matrix is used by all the machine learning algorithms and we use it as an algorithm in this project to detect the plagirasim in the source codes.By using the array matrix the arrays are represented using the matrix which is a 2-dimensional array. [2]

	Col1	Col2	Col3	Col4	....
Row1	Arr[0][0]	Arr[0][1]	Arr[0][2]	Arr[0][3]	
Row2	Arr[1][0]	Arr[1][1]	Arr[1][2]	Arr[1][3]	
Row3	Arr[2][0]	Arr[2][1]	Arr[2][2]	Arr[2][3]	
Row4	Arr[3][0]	Arr[3][1]	Arr[3][2]	Arr[3][3]	
⋮					

B. SKlearn Cosine Similarity

As a distance measurement metric between two points in the plane, cosine similarity is the cosine of the angle formed by two vectors. The cosine similarity metric is based solely on the cosine principle, which states that as distance increases, data point similarity decreases. The similarity between two vectors is gauged using the cosine similarity metric. In particular, it ignores variations in the magnitude or scale of the vectors and compares the similarity in their direction or orientation.

When it comes to machine learning, cosine similarity can be utilised for a variety of classification data and can be used to assist us identify the nearest neighbours when employed as an evaluation measure in the KNN method. Cosine similarity finds its main application for character types of data.

$$A \cdot B = \sum_{i=1}^{n} A_i B_i$$

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

C. Processing Steps

1. Dataset: To detect plagiarism in source codes, a dataset with source codes is collected, and the data is in.csv format. The information provided to us consists of source codes.

2. Data Preparation: We may name each attribute because the dataset we gathered cannot be applied directly to the detection algorithms.

3. Data Preprocessing: Predictive models' most crucial stage is data preparation since unclean data contains ambiguities, mistakes, redundancies, and transformations that must be removed.

4. Data Extraction: The properties are noted for the classification procedure.

5. Decision: We can determine whether the source codes contains plagiarism or not and how much of percentage it is plagiarized using data extraction and algorithms.

Experimental Results

To tell about the cosine similarity we use the following code.

```
from sklearn.metrics.pairwise import cosine_similarity
import numpy as np
array_vec_1 = np.array([[12,41,60,11,21]])
array_vec_2 = np.array([[40,11,4,11,14]])
print(cosine_similarity(array_vec_1, array_vec_2))
```

```
[[0.45227058]]
```

The following is the code for the detection of the plagiarism in a source code.

```
import os
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
from pathlib import Path

vectorize = lambda Text: TfidfVectorizer().fit_transform(Text).toarray()
similarity = lambda doc1, doc2: cosine_similarity([doc1, doc2])



def check_plagiarism(folder):

    student_files = [folder+"\\"+doc for doc in os.listdir(folder) if doc.endswith('.txt')]

    student_notes = []
    for File in student_files:
        student_notes.append(open(File).read())

    vectors = vectorize(student_notes)
    s_vectors = list(zip(student_files, vectors))

    plagiarism_results = set()
    for student_a, text_vector_a in s_vectors:
        new_vectors = s_vectors.copy()
        current_index = new_vectors.index((student_a, text_vector_a))
        del new_vectors[current_index]
        for student_b, text_vector_b in new_vectors:
            sim_score = similarity(text_vector_a, text_vector_b)[0][1]
            student_pair = sorted((student_a, student_b))
            score = (Path(student_pair[0]).name, Path(student_pair[1]).name, float(str(sim_score*100)[:5]))
            plagiarism_results.add(score)
    return plagiarism_results
```

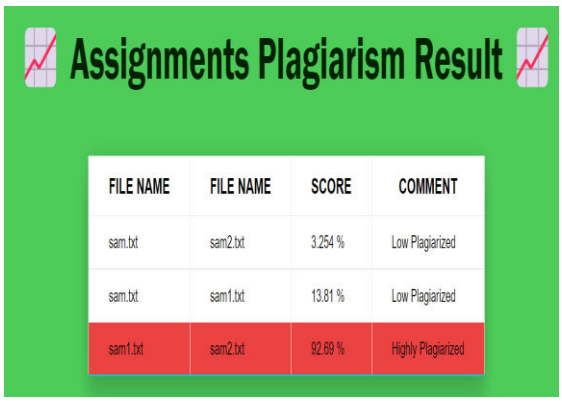
Now after applying the algorithms by using the flask we connect the machine learning code and the python code.here we uses the html and css to create a web page and it show how much percentage the given code is plagiarized and we divide the plagiarism into three types.

i.) Low plagiarised

ii.) Medium plagiarised

iii.)High plagiarised

Let us see the results.



By this the detection of plagiarism is done.

Conclusion and Future Work

To sum up, it's critical to realise that because plagiarism is a deliberate act on the part of the perpetrator, it is a problem of each individual's morale. There are many temptations to speed up work by utilising someone else's concept, and even if most of the time the odds are not in our favour, there is a fine line to walk between breaking the law and not. Unfortunately, plagiarism often occurs because early schooling did not provide adequate ethical instruction. Educating pupils to be independent and able to interact with society is just as vital as combating plagiarism.Here we use the best models to detect the plagiarism. We plan to improve our ability to detect the plagiarism in an easy and an efficient way.It helps the students to know about their mistakes and to rectify them in the future. So at last we conclude that the students have to write the source codes by using their knowledge and not to copy their codes from internet or by other ways.

References

[1] Nishesh Awale, Mitesh Pandey, Anish Dulal, Bibek Timsina, "Plagiarism Detection In Programming assignments Using Machine Learning", 2020, DOI: 10.36548/jaicn.2020.3.005.

[2] Farah Khaled, Mohammed Sabbih H. Al-Tamimi, "Plagiarism Detection Methods and Tools", Iraqi Journal of Science, 2021, Vol. 62, No. 8, pp: 2771-2783 DOI: 10.24996/ijs.2021.62.8.30.

[3] Hayden Cheers, Yuqing Lin, Shamus P.Smith, "Academic Source Code Plagiarism Detection by Measuring ProgramBehaviorSimilarity",IEEE,2021, DOI: 10.1109/ACCESS.2021.3069367.

[4] Simon, T. Myers, D. Hardy and R. Mason, "Variations on a theme: Academic integrity and program code", *Proc. 21st Australas. Comput. Educ. Conf. (ACE)*, pp. 56-63, 2019, DOI: 10.1145/3286967.

[5] E. Bogomolov, V. Kovalenko, A. Bacchelli and T. Bryksin, 2020, "Authorship attribution of source code: A language-agnostic approach and applicability in software engineering". arXiv preprint arXiv:2002.11593.

[6] Rodrigo C Aniceto, Maristela Holanda, Carla Castanho, Dilma Da Silva, "Source code plagiarism detection in an educational context: A literarture mapping", IEEE, 2021, DOI: 10.1109/FIE49875.2021.9637155.

[7] Xiao Li, Xiao Jing Zhong, "The source code plagiarism detection using AST", IEEE, 2010, DOI: 10.1109/IPTC.2010.90.

[8] Tapan P Gondaliya , Hiren Joshi, Hardik Joshi, "Source code plagiarism detection SCPDet : A Review", 2014, DOI: 10.5120/18471-9897.

[9] C. Arwin and S.M.M. Tahaghoghi, "Plagiarism detection across programming languages," Proceedings of the 29th Australasian Computer Science Conference-Volume 48, 2006, p. 286.

[10] S. Engels, V. Lakshmanan, and M. Craig, "Plagiarism detection using feature-based neural networks," Proceedings of the 38th SIGCSE technical symposium on Computer science education, 2007, p. 38.

[11] T Rahmat, A Ismail and S. Aliman, "Chest X-Ray Image Classification Using Faster R-Cnn", Malaysian J Comput., vol. 4, no. 1, pp. 225-36, 2019.

[12] J Rubin, D Sanghavi, C Zhao, K Lee, A Qadir and M.+ Xu- Wilson,"Large scale automated reading of frontal and lateral chest x-rays using dual convolutional neural networks", 2018.

[13] R Jain, P Nagrath, G Kataria, V Sirish Kaushik and D. Jude Hemanth,"Pneumonia detection in chest X-ray images using convolutional neural networks and transfer learning", Meas J

Int Meas Confed., vol. 165, pp. 108046, 2020.
[14] AK Jaiswal, P Tiwari, S Kumar, D Gupta, A Khanna and JJPC Rodrigues, "Identifying pneumonia in chest X-rays: A deep learning approach", Meas J Int Meas Confed, vol. 145, pp. 511-8, 2019.
[15] T. Nagamani.,et.al., "Malaria Detection Using Convolutional Neural Network", Journal of Engineering Sciences, ISSN NO: 0377-9254,Vol 11, Issue 6 , June, 2020..
[16] Butt et al., "Deep learning system to screen coronavirus disease2019 pneumonia", Applied Intelligence, vol. 1, 2020.
[17] Zech et al., "Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study", PLoS medicine, vol. 15, no. 11, pp. e1002683, 2018
[18] Asnaoui et al., "Automated methods for detection and classification pneumonia based on x-ray images using deep learning", arXiv preprint, 2020.
[19] Xianghong Gu et al., "Classification of Bacterial and Viral Childhood Pneumonia Using Deep Learning inChest Radiography", Proceedings of the 3rd International Conference on Multimedia and Image Processing (ICMIP 2018), pp. 88-93, 2018.
[20] Ioannis D. Apostolopoulos and Tzani A. Mpesiana, "Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks", Physical and Engineering Sciences in Medicine, pp. 1, 2020.
[21] Ashok Reddy Kandula, et al. "Machine Learning Algorithms for Classification of Genetic Mutations arose during the cell's rehabilitation to Cancer Tumor", Solid State Technology 64(2) 2752-2758, 2021.
[22] Baburao Markapudi, Kavitha Chaduvula, D.N.V.S.L.S. Indira & Meduri V. N. S. S. R. K. Sai Somayajulu, "Content-based video recommendation system (CBVRS): a novel approach to predict videos using multilayer feed forward neural network and Monte Carlo sampling method", published in the journal of Multimedia Tools and Applications (Springer Nature), published on 11th August 2022. PP:1-27, (2021).
[23] D.N.V.S.L.S.Indira, Babu Rao Markapudi, Kavitha Chaduvula, Rathna Jyothi Chaduvula, "Visual and buying sequence features-based product image recommendation using optimization based deep residual network", published in the journal of Gene Expression Patterns, Volume 45, 2022, September 2022.
[24] Dr GVSNRV Prasad, "Adaptive Optimization-enabled Neural Networks to handle the imbalance churn data in churn prediction", International Journal of Computational Intelligence and Applications,Dec-2011.
[25] Optimized Design of Hardware Monitors for Better Performance of Network Processors", International Journal of Research in Information Technology", Vol.4, No 1, June2017 ISSN2349-6002.
[26] "DE duplication Verification on Multimedia Data Stored in Cloud", International Journal for Scientific Research & Development", Vol.4, No1, May 2016 ISSN 2321-0613.
[27] "An Effective Secure Authorized Deduplication in Hybrid Cloud", International Journal of Science Engineering and Advance Technology",Vol.3, No1, Oct 2015 ISSN 2321-6905.
[28] "Secured Packet Hiding Technique for Packet Jamming Attacks", International Journal of Computer Trends and Technology",Vol.6,No1, Dec2013ISSN2231-2803.