SOFTWARE PLAGIARISM DETECTION IN MULTITHREADING USING MACHINE LEARNING

A PROJECT REPORT

Submitted By

ANJU SREEKUMAR

TVE18MCA009

to

the APJ Abdul Kalam Technological University

in partial fulfilment of the requirements for the award of the degree

of

Master of Computer Applications

Department of Computer Applications

College of Engineering

Trivandrum-695016

JULY 2021

Declaration

I undersigned hereby declare that the project report titled **"Software Plagiarism Detection in Multithreading Using Machine Learning"** submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University,Kerala is a bonafide work done by me under supervision of Dr. Sabitha S,HOD.This submission represents my ideas in my words and where ideas or words of others have been included.I have adequately and accurately cited and referenced the original sources.I also declare that I have adhered to ethics of academic honesty and integrity as directed in the ethics policy of the college and have not misrepresented or fabricated any data or idea or fact or source in my submission.I understand that any violation of the above will be a cause for disciplinary action by the Institute and/or University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained.This report has not been previously formed the basis for the award of any degree, diploma or similar title.

Place : Trivandrum

Date : 27/06/2021

Anju Sreekumar

DEPARTMENT OF COMPUTER APPLICATIONS

COLLEGE OF ENGINEERING
TRIVANDRUM

CERTIFICATE

This is to certify that the report entitled **Software Plagiarism Detection in Multi-threading Using Machine Learning** submitted by **Anju Sreekumar** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications is a bonafide record of the project work carried out by her under my guidance and supervision.This report in any form has not been submitted to any University or Institute for any purpose.

Internal Supervisor External Supervisor

Head of the Dept

Acknowledgement

First and for most I thank **GOD** almighty and to my parents for the success of this project.I owe a sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I am extremely thankful to **Dr. Jiji C V**,Principal,College of Engineering Trivandrum for providing me with the best facilities and atmosphere which was necessary for the successful completion of this project.

I am extremely grateful to **Dr. Sabitha S**,HOD,Dept of Computer Applications,for providing me with best facilities and atmosphere for the creative work guidance and encouragement.

I express our sincere thanks to **Dr. Sabitha S**,HOD,Department of Computer Applications,College of Engineering Trivandrum for her valuable guidance,support and advice that aided in the successful completion of my project.

I profusely thank other Asst.Professors in the department and all other staffs of CET,for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped me in the successful completion of this project.No words can express my humble gratitude to my beloved parents and relatives who have been guiding me in all walks of my journey.

Anju Sreekumar

Abstract

Software plagiarism,is an act of software theft severely affect both open source communities and honest software companies.The recent events include the lawsuit against Verizon by Free Software Foundation for distributing Busybox in its FIOS wireless routers,and for the violation of licensing terms of Joltid Skype's VOIP service are almost terminated.Unfortunately software plagiarism is difficult to detect but easy to implement.Today software birthmark approaches are available,are applicable only to sequential programs.Existing plagiarism techniques are no longer applicable to modern software with multiple threads.This project proposed a method for software plagiarism detection in multi-programming languages based on machine learning approaches.Software birthmarks is a method for finding the detection of programs that may have been taken by measuring the similarity between the two programs.The project focuses on birthmark generation,similarity calculation and plagiarism detection.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Software plagiarism,is an act of illegally copying others' code,affect both open source communities and honest software companies.Open source software allows its usage and modification under certain types of licenses.For example,GPL (GNU General Public License) allows users to modify GPL programs freely only when they agrees the tenets of GPL.However,for commercial interests,some companies and individuals incorporate third party software for violating the licensing terms.These intentional or unintentional software license violations lead to serious controversies from time to time.For example,licensing debate between Skype and Joltid that almost terminated Skypes voice-over-IP service.

Software plagiarism detection techniques are,therefore,welcomed by programmers and who want to protect their code and companies.So they can avoid costly lawsuits.As multithreaded programs become very popular,plagiarism of multithreaded programs starts to plague the software industry.The technique for detecting plagiarism in software has come a long way.Existing dynamic birthmark methods can only be used with sequential programs.This problem can be easily solved by an software plagiarism detection in multithreading that ensures integrity and faster result.

Chapter 2

Problem Definition and Motivation

Software plagiarism detection in multithreading can be defined as the process to determine whether a defendant(software suspected of plagiarism)software contains plagiarism code by comparing it to a plaintiff software(original software).The process of software plagiarism detection could be useful for both programmers and companies,since it encourage brilliant innovations and direct our future research efforts.It is usually treated as a supervised machine learning problem where feature extraction plays an important role.Through my project I'm trying to develop a machine learning model which can be used to detect software plagiarism.

The major motivation behind choosing the project was the drawbacks of the existing system.The system that we have now is dynamic birthmark approaches.The major drawbacks of the existing system are mentioned below,

- The dynamic birthmark approaches are only applicable for sequential programs.
- The dynamic birthmark approaches are no longer applicable to modern software with multiple threads.
- With the help of various powerful automated semantics-preserving code unclear tools,plagiarists are able to make significant changes to code without changing its functionality.

The major objective of the project is to automatically detect the software plagiarism through the project and try to save from software thefts for future research efforts and to encourage innovations.Through this project I can assure that the project can achieve it.

Chapter 3

Literature Review

The software plagiarism is a hot problem since ages.Also different detection techniques are arise in this period.As a part of my literature review I went through various publications and talks on this topic.The quick summary of my findings are specified in this chapter.

3.1 Using Software Watermark Techniques

Collberg and Thoborson proposed software watermark techniques.Software watermark is a unique identifier inserted into the software,which is easy to verify but hard to remove.It is basically to embed secret information which identifies the program author.To completely prove software theft,watermark has to be embedded into all related modules in advance.This becomes difficult when the number of modules is large.

3.2 Using Software Birthmark

Another approach is software birthmark to solve software plagiarism.A software birthmark is a unique characteristic that a program essentially possesses,which is used to identify the program.The idea is that if two programs have the same birthmark then it is highly possible that one is a copy of the other.Birthmarks are classified into two types.Static software birthmark and dynamic software birthmark.Static birthmarks are the static behavior of a program at compile time.Dynamic birthmarks are the dynamic behavior of a program at run time.

3.3 Using DYKIS Technique

In 2015,the paper "Software Plagiarism Detection with Birthmarks Based on Dynamic Key Instruction Sequences" unveiled the DYKIS approach.They pointed that matching the software content need less importance so importance is given to logic.The range of similarity score is between 0 to 1 and is calculated using jaccard method.

3.4 Using Fuzzy Hashing Technique

A new method that is fuzzy hashing method is presented in the paper "Revisiting the Opportunities and Challenges in Software Plagiarism Detection".It is the better method for detection.This paper is presented in 2020.Using fuzzy hashing method it convert a birthmark gained by a older method to a hash function.Then compute similarity based on this hash value.Cryptographic hash functions such as SHA1 or MD5 cannot be used to compute similarity for this process.To compute similarity,fuzzy hashing also known as context-triggered piece wise hashing (CTPH),which was developed as a method for identifying sufficiently different data strings.To apply this method the software is first checked by the proposed birthmark method.Any discovered possible copies are then investigated in more detail using the traditional birthmark or other methods.

By concluding all the data,I can say that the problem of software plagiarism detection can be done using various techniques.Among them one of the best method is to detect software plagiarism using software birthmark technique also with fuzzy hashing and then develop a machine learning model.Through this project got to know that the result of this model will be quite encouraging and the margin of error will be minimum.

Chapter 4

Requirement Analysis

4.1 Purpose

The detection of software plagiarism using current techniques are time consuming and are only for sequential programs.The purpose of this project is to develop a system to detect software plagiarism for multi threaded programs.The "Software Plagiarism Detection in Multi-threading" is build by using Machine Learning.

4.2 Overall Description

Software plagiarism is an act of reusing someone else's code,in part or whole.It is an act of theft.Recently software theft has become a very serious concern to software companies and open source communities.Software birthmark techniques are popular software protection technique.Software birthmark technique can be defined as a mechanism which identifies the unique characteristics of a software.Depending on its extraction relies on program runs,a software birthmark can be either considered as static or dynamic.In the current system developed,a model is created that is trained by machine learning to detect software plagiarism written by the user.Features are extracted using birthmark generation algorithm.Then software birthmark are generated and similarity is found using fuzzy hash technique.

4.2.1 Hardware Requirements

- Processor : Intel Core i3
- Storage : 512 GB Hard Disk space
- Memory : 4 GB RAM

4.2.2 Software Requirements

- Operating System : Linux/Windows
- Platform : Visual Studio
- Front end : Visual C#

4.3 Functional Requirements

The functional requirements represent the deliberated behaviour of the system.The proposed system consists of four modules.It includes

- **Birthmark generation :** The two software which are to be checked has to run in the background.Each one of them will be monitored for 30 seconds each and details such as CPU,memory and disk usage are collected.Gathering details such as packet used,memory usage,CPU,disk,status of ports using system calls.A port number is thus ranging from 0 to 65535.
- **Thread generation :** A process is a program that is currently running.A thread is a process's smallest executable unit.A process can have multiple threads.Multi-threading is a type of execution model that allows multiple threads to exist within the context of a process such that they execute independently but share their process resources.Details such as Thread ID,priority,privilege and so on are noted.
- **Comparison of the plaintiff and defendant program :** Here compare the plaintiff(program suspected of plagiarism)and defendant(original program)program using 40 threads collected with each.Threads are collected based on priority of the process.
- **Applying fuzzy hashing technique :** A hash function is a function that can be used to convert data of arbitrary size onto data of a fixed size.Cryptographic hash functions

have a disadvantage as just one bit creates very different results that are useless when checking similarity.Hence non cryptographic hash function such as fuzzy hashing is used for similarity purposes.

4.4 Non Functional Requirements

4.4.1 Performance Requirements

- Accuracy : Accuracy in functioning and the nature of user-friendly should be maintained by the system.
- Speed : The system should be able to provide speed.
- Low cost: This system is very cheap to implement and is also user-friendly.
- Less Time consuming: It uses very less time comparing to the existing system .
- User Friendly: This proposed system is highly user friendly they enables to create a good environment.

4.4.2 Quality Requirements

- Scalability : All of the functional requirements will be met by the program.
- Maintainability : The system should be maintainable.It should keep backups to atone for system failures,and should log its activities periodically.
- Reliability : The acceptable threshold for down-time should be large as possible.i.e. mean time between failures should be large as possible.If the system is broken,time required to get the system backup again should be minimum.
- Availability: This system is easily available as the core equipment's in building the software is easily obtained.
- High- Functionality: This system is highly functional in all environment since,They are highly adaptable.

Chapter 5

Design And Implementation

The proposed system is used to find software plagiarism using a software birthmark technique.For software birth mark features are extracted using feature extraction algorithm.

5.1 Overall Design

The proposed system is a simple software based on machine learning.The client part and server part is used by the admin,because of this project is software.A single file comparison also achieved.The input is passed to server and the evaluated result is given back to the admin.It is developed in visual C#.

5.1.1 System Design

The system is software based.The inputs are plaintiff software(original software) and defendant(software suspected of plagiarism) software.The input is taken from the user through software and the input is passed to the visual c# program running in the server side.The server program perform tasks such as processing system calls,feature extraction,birthmark generation,similarity matching and detection on the input data.

For feature extraction feature extraction algorithm is used.Hence for birthmark generation TOB(Thread oblivious Birthmark) is used.

5.1.2 Methodology

There are two parts in this project.The first part is the creation of the model and the second one is the creation of user program which will work with the pre-trained model.

The main process of the software plagiarism detection in multithreading is the creation of the trained model.The major steps in the model creation are Feature extraction,birthmark creation,similarity matching and detection.The major steps in the model creation are detailed below.

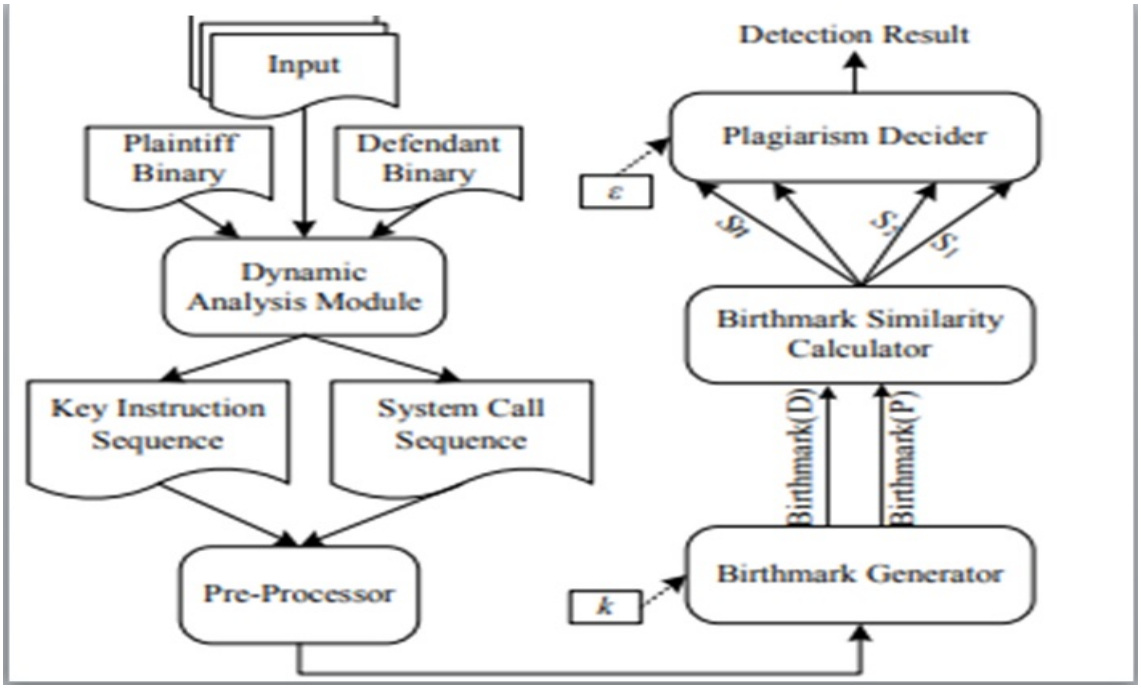

Figure 5.1: Architecture of model Creation

- **feature extraction:**Feature extraction is the most important part of any machine learning task and so is the case with me.Here the features of the software are extracted using feature extraction algorithm.Activities of all ports are listening in this phase.

- **Birthmark Generation:**Thread oblivious birthmark (TOB) is the technique used for the generation of the birthmark.It extracts the features collected in the preprocessing stage.The collected unique features are then used for similarity calculation.

- **Similarity Matching:** In this phase,the typical range of similarities is 1 to 100.0 means two software are completely different,similarity greater than 95 means the one software is strongly suspected a copy of the other.The results are recorded for the next stage which is the model plagiarism detection.

- **Plagiarism detection:** The results of the similarity matching data is used for the plagiarism detection.If the value is higher than the threshold value then plagiarism is detected otherwise not.

The second part of the project is to build the user interface.The user interface is build using Visual studio.The input is fed into the server through this.The results returned are also displayed.The interface is built in a way such that it is easy and understandable for the person who uses it.For that I use visual C# to provide the better user experience.

5.2 Data Flow Diagram

DFD is one of the graphical representation techniques used in a project to show the flow of the data through a project.DFD helps us to obtain an idea about the input,output,and process involved.The things absent in a DFD are control flow,decision rules,and loops.It can be described as a representation of functions,processes that capture,manipulate,store,and distribute data between a system and the surrounding and between the components of the system.The visual representation helps for good communication.

It shows the journey of the data and how will it be stored in the last.It does not provide details about the process timings or if the process shall have a parallel or sequential operation.It is very different from a traditional flow chart or a UML that shows the control flow or the data flow.

In level 0 the basic data flow of the application is showcased.It does not show the flow of data much deeper.It will be evaluated in the higher levels of Data Flow Diagram.The Data Flow Diagram of Automated essay scoring system is shown below.

Figure 5.2: Level 0 DFD

The diagram shows Level 0 Data flow diagram of the Software plagiarism detection in multi-threading.As the diagram indicates there is only an admin part.The input of the project is the two software by the admin and which is given to the software plagiarism detection system.The value of the evaluation is passed back to the admin through the application.Since there is no database in the application,the data is not stored anywhere.The data is lost after the evaluation.

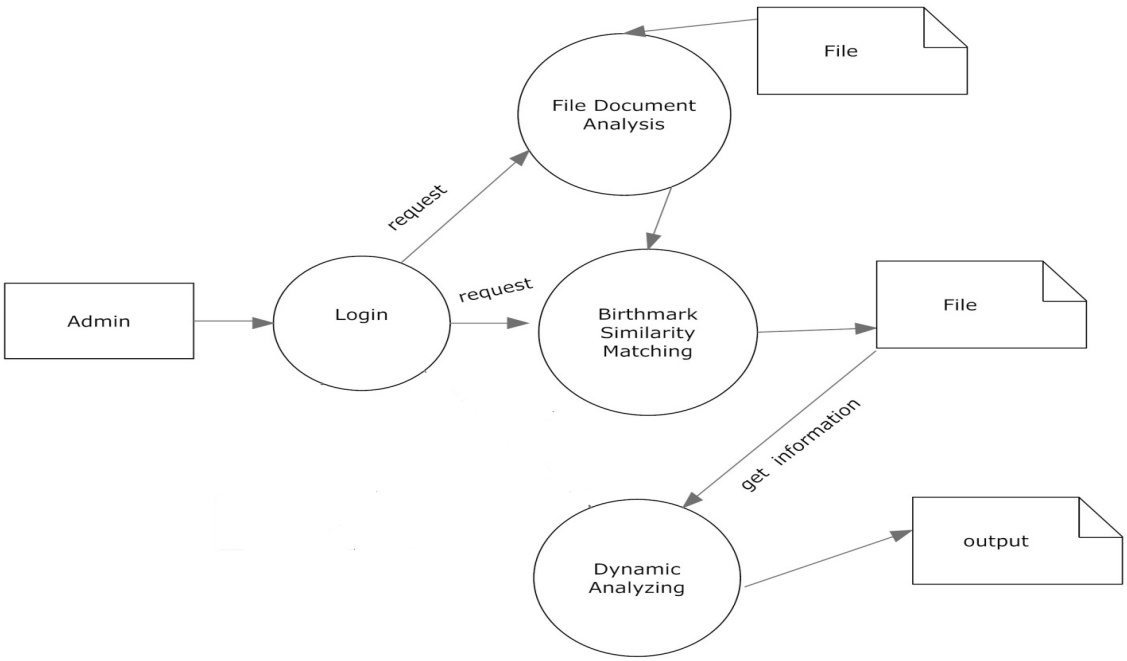

Figure 5.3: Level 1 DFD

5.3 Screenshots

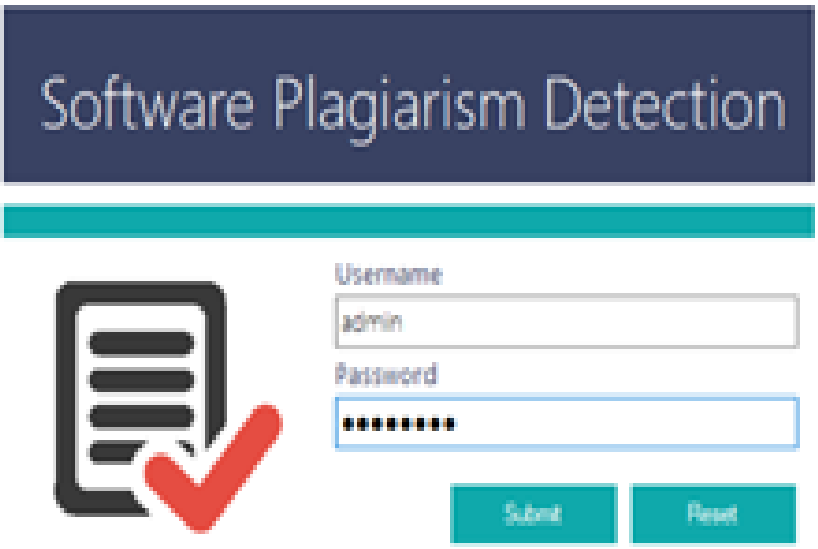

Figure 5.4: Login page

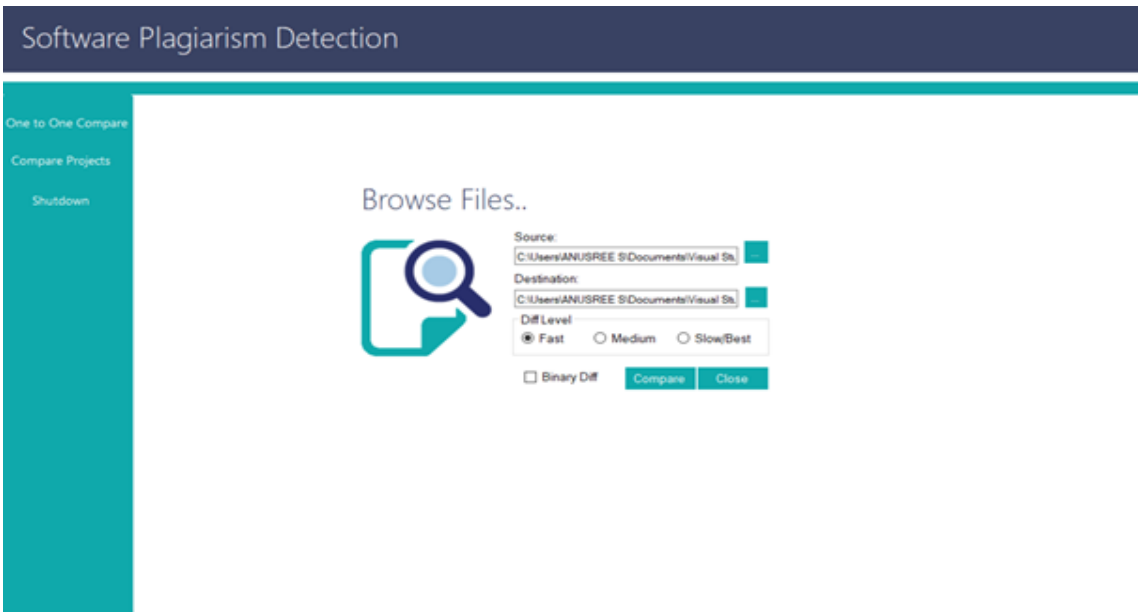

Figure 5.5: Browse files for file comparison

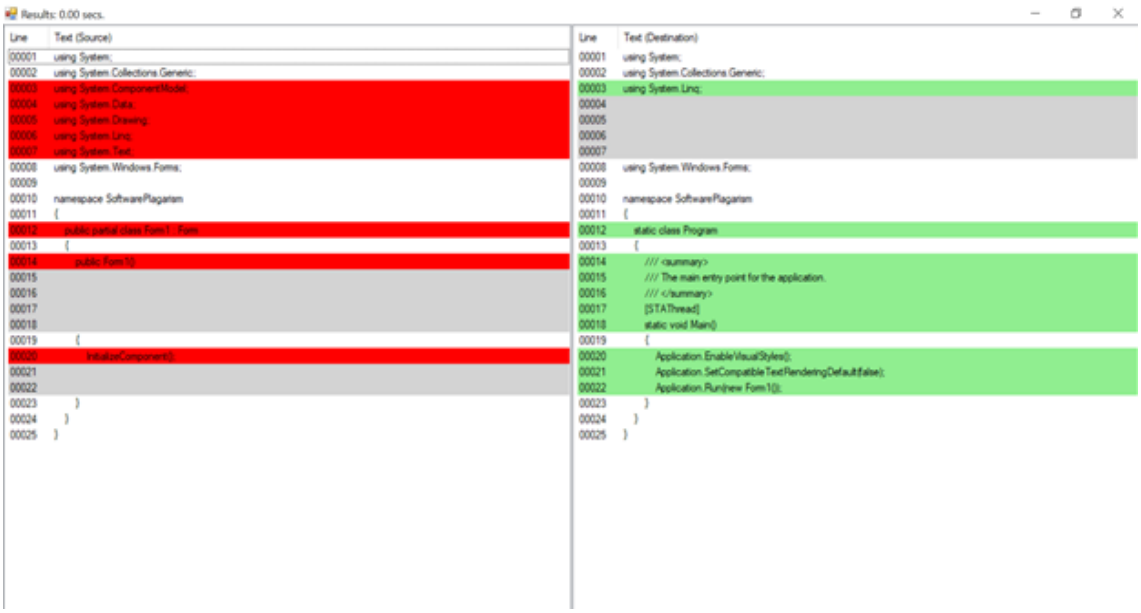

Figure 5.6: Result of file comparison

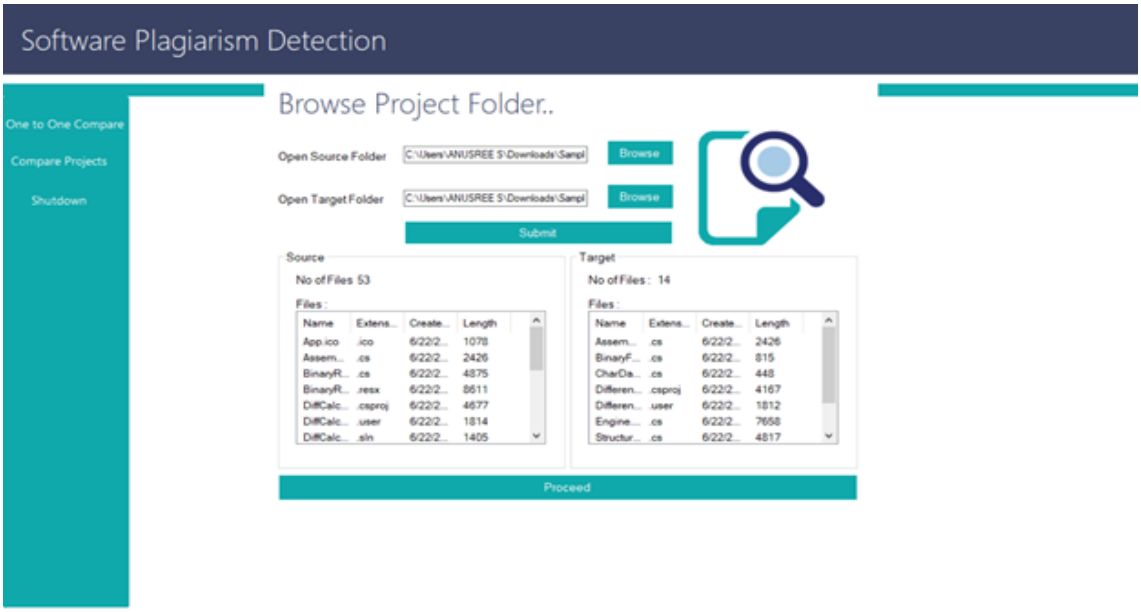

Figure 5.7: Input for software comparison

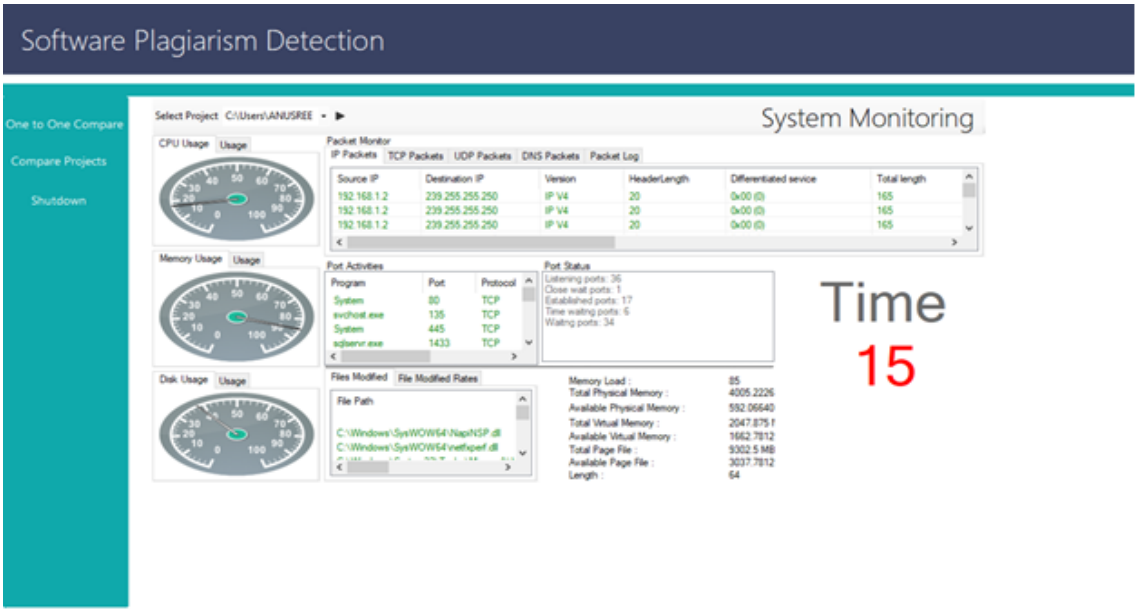

Figure 5.8: System monitoring

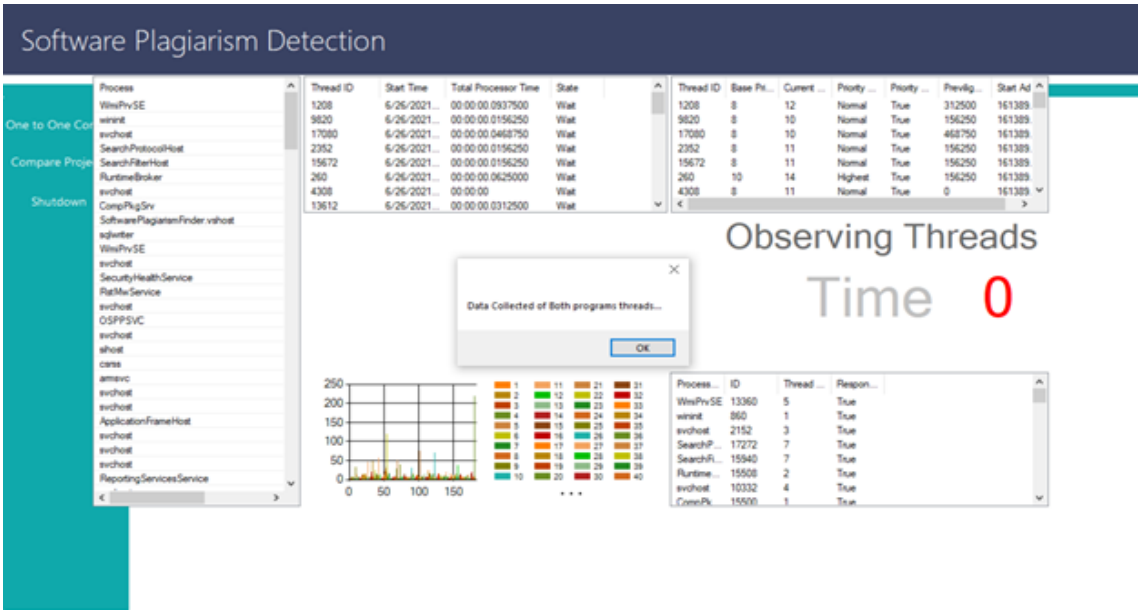

Figure 5.9: Observing threads

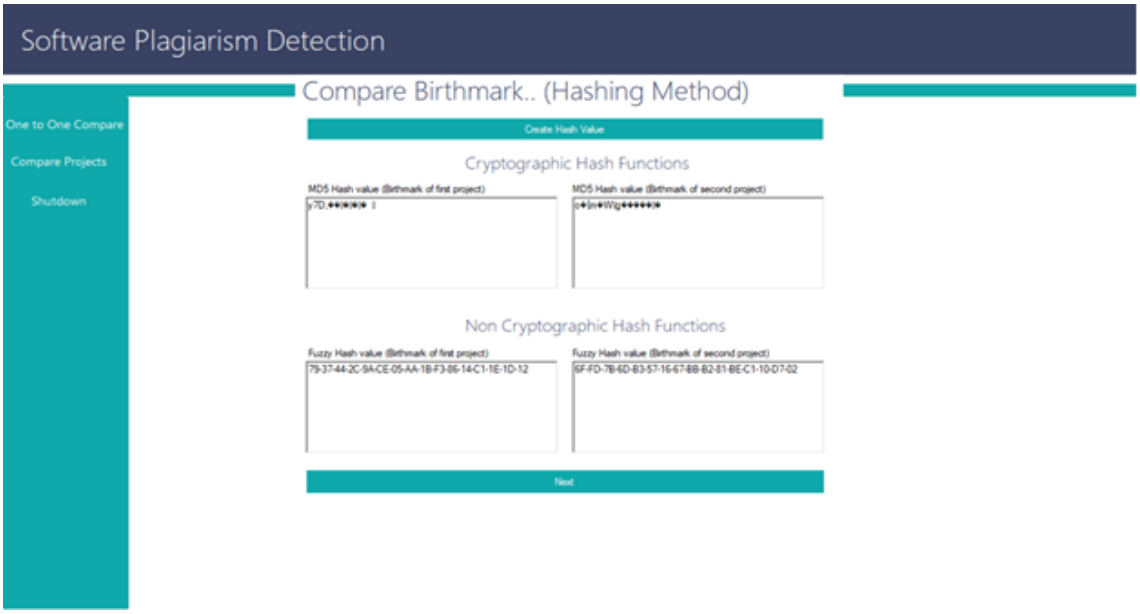

Figure 5.10: Applying hash functions

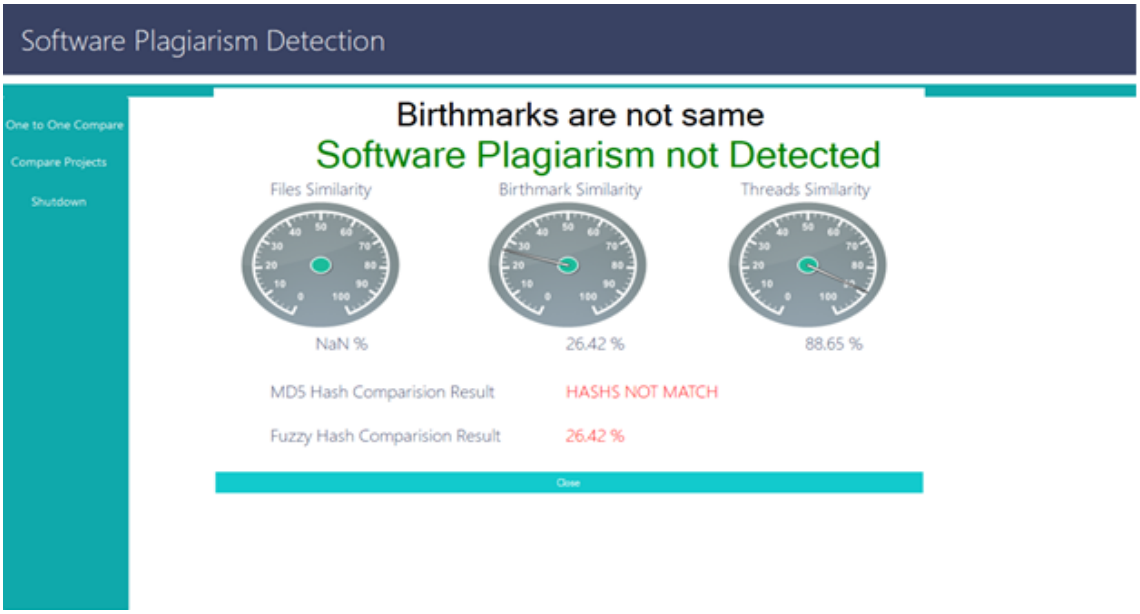

Figure 5.11: Plagiarism detection

Chapter 6

Coding

Algorithm 1 Algorithm for birthmark generation:

1: Normalization is one of the basic steps in text mining and text processing.Punctuation and stop words are removed in the normalization process.

2: Turn text to set of clauses.Suspicious document and the reference document are divided to a series of sentences.Each sentence of the mistrusted document will be compared to all the sentences of the original document.In this step to reduce run time,filtering will be done.

3: For creating corresponding graph each sentence will be converted to a graph.The nodes of the graph are unique words.In this graph,an edge will be created between a particular word and 4 words before and after it in the document.

4: When graph creation is complete,then looking for nodes in the suspected document that is common with the node of the reference document.This method based on simple idea indicates that the two graphs are similar when they have similar nodes.So use this method for our specific graph.

5: Integrate plagiarism labeled Sentences.In this step,integrate sentences with plagiarism label(output of step 4 of the algorithm)based on start and end offset of sentences in text.If no labeled sentences there exist,then we can assume that no plagiarism is occurred.

Algorithm 2 Algorithm for plagiarism detection:

1: Read the input software through the user interface.

2: Then access the input software and perform the preprocessing tasks on it.

3: The preprocessed software is used to extract the required features for birthmark creation using birthmark creation algorithm.

4: Using the birthmark generation algorithm,unique features are collected.

5: To improve efficiency and speed fuzzy hash technique then applied to detect plagiarism.

6: The result will be displayed in the system.

Chapter 7

Testing

7.1 Testing and various types of testing used.

Once a software is developed,the major activity is to test whether the actual results match with the experimental results.This process is called testing.It's used to make sure that the developed system is defect free.The main aim of testing is to find the errors and missing operations by executing the program.It also ensure that all of the objectives of the project are met by the developer.The objective of testing is not only to evaluate the bugs in the created software but also finding the ways to improve the efficiency,usability and accuracy of it.It aims to measure the functionality,specification and performance of a software program.Tests are performed on the created software and their results are compared with the expected documentation.When there are too much errors occurred,debugging is performed.The result after debugging is tested again to make sure that the software is error free.The major testing processes applied to this project are unit testing,integration testing and system testing.In unit testing,my aim is to test all individual units of the software.It makes sure that all of the units of the software works as it intended.In integration testing,the combined individual units are tested to check whether it met the intended function or not.It helps me to find out the faults that may arise when the units are combined.In system testing the entire software is tested to make sure that it satisfies all of the requirements.The tables shown below describes the testing process occurred during the development of this project "Software plagiarism detection in multithreading using machine learning".This defines the various steps took to create the project error free.

7.1.1 Unit Testing

Text Cases and Result

Sl No	Procedures	Expected result	Actual result	Pass or Fail
1	Upload the source and destination program from specified location	Load the project	Same as expected	Pass
2	Check whether a port is alive or not	Reply from the port	Same as expected	Pass
3	Generate multi-threaded class	Check thread is live if not create it	Same as expected	Pass
4	Compare different birthmark technique	Compare the all the valid parameters in the graph	Same as expected	Pass

Table 7.1: Unit test cases and results

7.1.2 Integration Testing

Text Cases and Result

Sl No	Procedures	Expected result	Actual result	Pass or Fail
1	Process the input file from the source and destination folder	Check plagiarism from the file and give result	Same as expected	Pass
2	Generate Dynamic Birthmark Generation class	Display the result of Multithread Dynamic Birthmark Class	Same as expected	Pass
3	Graphical Analysis of different method	Compare the result and gives graphical output	Same as expected	Pass

Table 7.2: Integration cases and result

7.1.3 System Testing

Text Cases and Result

Sl No	Procedures	Expected result	Actual result	Pass or Fail
1	To run server	Server program executed successfully, hence the entire program worked without any crash	Same as expected	Pass
2	Plagiarism detection	Result will be Plagiarism detected percentage	Same as expected	Pass

Table 7.3: System test cases and results

Chapter 8

Results and Discussion

The main aim of the project was to compare and detect software plagiarism using a machine learning model.It is observed that the system performs all the functionalities as expected.By using this machine learning model the computer can detect plagiarism of two software at a time.

8.1 Advantages and Limitations

The proposed system is a machine learning model to evaluate the input software and find whether plagiarism or not.The proposed system posses more advantages over the existing system.The proposed system save a huge amount of time.Like every other system,this system also have it's own disadvantages.But they are negligible while comparing with the advantages and they can be overcame in future.

8.1.1 Advantages

- Can save the time needed for the detection of software plagiarism.This time can be spend more productive by the developers.
- Reduces chances of confusions regarding the original and duplicate software.
- Improve efficiency and gives the live result.
- Software plagiarism related issues can be reduced.

8.1.2 Limitations

- Only two software can be compared at a time.That is lack of real world plagiarism cases.Hence the results are neither too good nor too bad.
- The system also suffer the same limitation of dynamic birthmarks in exhaustively covering all behaviors of a program.

Chapter 9

Conclusion and Future Scope

The act of illegally copying others code is refers to as software plagiarism.Several techniques have been developed to impede this.In which software birthmark based on system call is considered in this project work.System calls are believed to be a fundamental run time indicator of program behavior.Hence it is a good candidate for behavior based birthmarks.The trend towards multithreaded programs is creating a gap between the current software development practice and the software plagiarism detection technology,as the existing dynamic approaches are only for sequential programs.Hence it is also taken in to consideration in this work.

The software birthmark thus created is large in size and hence takes time to compare.As the size of program increases,the time taken to compare will again increase.To solve this,fuzzy hashing technique is introduced.The hash of birthmark is computed and then the hashes are compared to find similarity.The time taken to compare has tremendously decreased by introducing it.The proposed method also suffer the same limitation of dynamic birthmarks in exhaustively covering all behaviors of a program.

One way to reduce the problem is to combine with testing techniques.One problem of plagiarism detection researches face is the lack of real world plagiarism cases.In recent years,software program plagiarism on mobile markets starts to increase.Identifying pairs of applications that plagiarism may exist is extremely laborious.So this can be taken as one of the future work.

Bibliography

[1] Z. Tian, Q. Zheng, T. Liu, M. Fan, E. Zhuang, and Z. Yang, Software plagiarism detection with birthmarks based on dynamic key instruction sequences,IEEE Transactions on Software Engineering Dec. 2015.

[2] Z. Tian, Q. Zheng, T. Liu, M. Fan, X. Zhang, and Z. Yang, Plagiarism detection for multithreaded software based on thread-aware software birthmarks, 2014 International Conference on Program Comprehension.

[3] G. Myles and C. S. Collberg, Detecting software theft via whole program path birthmarks, 2004 International Conference on Information Security.

[4] Z. Tian, T. Liu, Q. Zheng, E. Zhuang, M. Fan, and Z. Yang, Reviving sequential program birthmarking for multithreaded software plagiarism detection, IEEE Transactions on Software Engineering 2017.