

SOFTWARE PLAGIARISM DETECTION IN MULTITHREADING USING MACHINE LEARNING

Presented by : Anju Sreekumar

S6 MCA

Roll No:08

TVE18MCA009

Guided by : Dr.Sabitha S

June 28, 2021

COLLEGE OF ENGINEERING TRIVANDRUM

OVERVIEW

- 1.INTRODUCTION
- 2.LITERATURE REVIEW
- 3.MOTIVATION
- 4.PROPOSED SYSTEM
- 5.EXPERIMENTAL EVALUATION
- 6.MODULES
- 7.MODEL
- 8.SCREEN SHOTS
- 9.RESULTS AND DISCUSSION
- 10.CONCLUSION
- 11.REFERENCE

INTRODUCTION

- The Software plagiarism, which arises the problem of software piracy is a growing major concern nowadays.
- The customers may develop a modified version of the original software in other types of programming languages.

- Semantics-preserving code obfuscations.
Developed by Lannan Luo, combines rigorous program semantics with longest common subsequence based fuzzy matching.
- Software watermarking.
Collberg and Thoborson proposed software watermark technique By embedding a unique identifier, i.e., a watermark.
- Software birthmarking.
Is illustrated by Myles and Collberg, birthmarking attempts to extract a set of characteristics that can be used to uniquely identify a program.

MOTIVATION

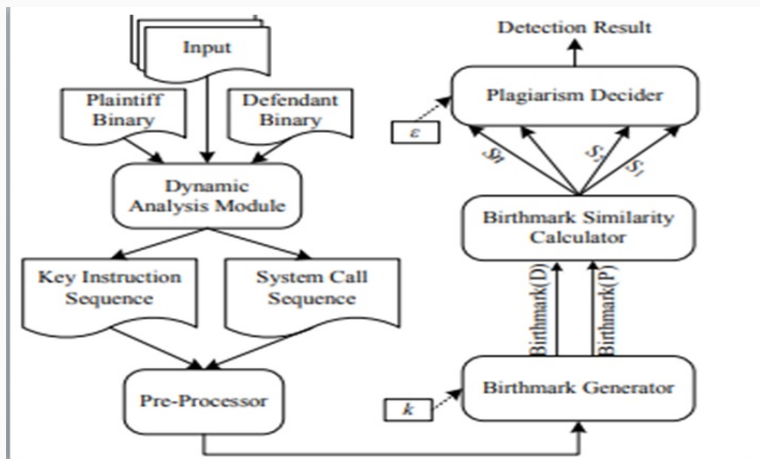
- The Software plagiarism, It is a serious risk to the software industry that gives huge economic damages every year.
- GPL (GNU General Public License) allows users to modify GPL compliance programs freely, as long as the derivative works also follow the tenets of GPL
- However, driven by commercial interests, some companies and individuals incorporate third party software without respecting the licensing terms.

- Proposed a methodology for software plagiarism detection in multi-programming languages based on machine learning approaches.
- Software birthmarks have been proposed as a method for enabling the detection of programs that may have been stolen by measuring the similarity between the two software.

- Hardware Requirements
 - Processor : Intel Core i3
 - Memory : 4 GB RAM
- Software Requirements
 - Operating System : Windows
 - IDE: Visual Studio

- Birthmark creation
- Similarity Calculation
- Plagiarism detection

MODEL



SCREEN SHOTS

Login page



The image shows a login page for 'Software Plagiarism Detection'. At the top, a dark blue header contains the title in white. Below this is a teal horizontal bar. On the left, there is a logo consisting of a black document icon with a red checkmark overlaid. To the right of the logo are two input fields: 'Username' with the text 'admin' and 'Password' with masked characters. Below these fields are two teal buttons labeled 'Submit' and 'Reset'.

Software Plagiarism Detection



Username
admin

Password
.....

Submit Reset

Result of file comparison

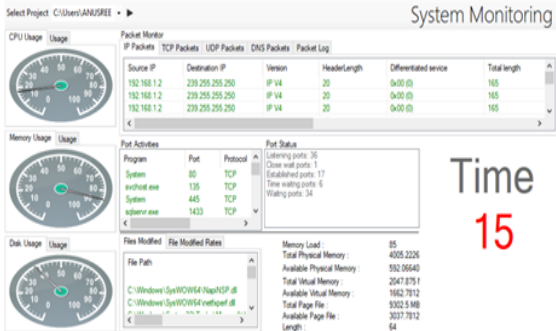
Results: 0.00 secs.

Line	Text (Source)	Line	Text (Destination)
00001	using System;	00001	using System;
00002	using System.Collections.Generic;	00002	using System.Collections.Generic;
00003	using System.ComponentModel;	00003	using System.Linq;
00004	using System.Data;	00004	
00005	using System.Drawing;	00005	
00006	using System.Linq;	00006	
00007	using System.Text;	00007	
00008	using System.Windows.Forms;	00008	using System.Windows.Forms;
00009		00009	
00010	namespace SoftwarePlagiarism	00010	namespace SoftwarePlagiarism
00011	{	00011	{
00012	public partial class Form1 : Form	00012	static class Program
00013	{	00013	{
00014	public Form1()	00014	/// <summary>
00015		00015	/// The main entry point for the application.
00016		00016	/// </summary>
00017		00017	[STAThread]
00018		00018	static void Main()
00019	{	00019	{
00020	InitializeComponent();	00020	Application.EnableVisualStyles();
00021		00021	Application.SetCompatibleTextRenderingDefault(false);
00022		00022	Application.Run(new Form1());
00023	}	00023	}
00024	}	00024	}
00025	}	00025	}

One to One Compare

Compare Projects

Shutdown



Time
15

Compare Birthmark.. (Hashing Method)

One to One Compare

Compare Projects

Shutdown

Create Hash Value

Cryptographic Hash Functions

MD5 Hash value (Birthmark of first project)

57D*****

MD5 Hash value (Birthmark of second project)

54j*****

Non Cryptographic Hash Functions

Fuzzy Hash value (Birthmark of first project)

79-37-44-2C-5A-CE-05-AA-1B-F3-06-14-C1-1E-1D-12

Fuzzy Hash value (Birthmark of second project)

6F-FD-7B-4D-83-57-16-67-8B-82-81-8E-C1-10-D7-02

Next

One to One Compare

Compare Projects

Shutdown

Birthmarks are not same

Software Plagiarism not Detected

Files Similarity



NaN %

Birthmark Similarity



26.42 %

Threads Similarity



88.65 %

MD5 Hash Comparison Result

HASHS NOT MATCH

Fuzzy Hash Comparison Result

26.42 %

Done

- Each of the features, functionalities are working correctly. This proposed system have met all the proposed requirements in the proposed system.
- Expected software plagiarism detected perfectly

- It can extend my work in the next higher level by enhancing our model to generate software plagiarism, comparing with real world plagiarism cases.
- Identifying pairs of applications that plagiarism may exist is extremely laborious. So this can be taken as one of the future work.

- The proposed system is for software plagiarism detection in multi-programming languages based on machine learning approaches.
- This is completely a Machine Learning project, which makes use of fuzzy hashing technique.



Xi Xu,Ming Fan "Revisiting the Challenges and Opportunities in Software Plagiarism Detection", IEEE May 31,2020

THANK YOU
