SOFTWARE PIRACY DETECTION USING DEEP LEARNING APPROACH

1Prof. Pritam Ahire, 2Mrunal Gaikwad, 3Pratibha Kasar, 4Sonal Bhatter, 5Yash Chikane
1Assistant Professor, 2Student, 3Student, 4Student, 5Student
1Computer Engineering,
1DY Patil Institute of Engineering, Pune, India

Abstract: In today’s world software piracy is high risk to compromise the security in computer world. The detection of software piracy is the main aim in the field of cyber security. In proposed system, a combined deep learning approach is proposed to identify and economical damages to the software industry. The traditional methods available may solve the concern but high computational cost will be needed to do so. The proposed system will try to detect software piracy by providing less computational cost will be needed to do so. The proposed system will try to detect software piracy by providing less computational cost to improve the accuracy. The deep learning approach involves two steps: first one being pre-processing. This will break the source code in small pieces for deep analysis, converting the code into some meaningful information and removing the noisy data. Tokenization is used to transform this clean data into some useful information. TF-IDF is used for weighting process i.e. to zoom the contribution of even token. The second step uses TensorFlow neural network to identify pirated software using source code plagiarism. TensorFlow has different types of layers which can be configured for complex computations, training the data. The in-depth learning approach is designed to identify similar source codes in different types of programming languages using TensorFlow framework. Then, the extracted similar codes are used to identify the pirated software.

Keywords – Deep Learning Approach, Tensor Flow, Neural Network, Plagiarism, TF-IDF, Tokenization, Normalization.

I. INTRODUCTION

Source code plagiarism detection in programming assignments is a task many higher education academics carry out. Source code plagiarism occurs when students reuse source code authored by someone else, either intentionally or unintentionally, and fail to adequately acknowledge the fact that the particular source code is not their own. Software piracy can be referred as illegally stealing citations. Currently, every other installed software is pirated. There are many scenarios of this happening, the attacker may crack the original legal software and reconstruct or re-design the logic into other programming language or may change minor details of the software. It is very exasperating to catch such assailants malicious activities as all the programming language have their own syntax and semantic structures.

Currently, Software piracy is high risk for security of software. It may cause reputational and economic damages. Now a day every other software is pirated there are many scenarios in which it can occur, the programmer may crack the original legal software and reconstruct or re-design the logic into other programming languages or may change the minor details of the software so we proposed a combine Deep learning approach to detect the pirated software. The TensorFlow Deep neural network is proposed to identify pirated the techniques like Tokenization and Weighting are used to filter noisy data. The dataset is collected from Google code jam (GJC) to find the software piracy. The process of software piracy is very exasperating to each such assaulter malicious activities as all the programming languages have their own syntax and semantic structure. The
experiment result shows that how much percentage of software code is plagiarized which be effective from current available methods.

II. RELATED WORK

The proposed system has three main module: Uploading data from GCJ(Google Jam Code), Software piracy analysis unit, piracy detection unit.

Uploading data from GCJ:-
The Required code is collected from Google code jam for detecting and analyzing software piracy.

Piracy analysis unit:-
To analyze software piracy following procedure steps are followed:
1. Preprocessing
2. Tokenization
3. Token’s weighting
4. Normalization

1. Preprocessing:
   This process is used to break the original code into member of pieces. It converts code into member meaningful information by revolving noisy data.

2. Tokenization:
In tokenization broken pieces of code obtained from pre-processing phase has transformed into pre-processing phase has transformed into useful tokens. Techniques such as stemming root word are used.

3. Token’s weighting:
Various weighting techniques can be used to zoom in the contribution of each token. TF-IDF techniques is used for weighting for tokens.

4. Normalization:
Normalization is used to eliminate redundancy and undesirable characteristics and to obtain values on a common standard scale.

**Piracy detection unit:**
- The tensorflow neural network is used to detect the software piracy.
- Similar code frequency is used to detect piracy.
- This unit shows the result of pirated software. In which it detects how much of percentage the software is pirated.

**Proposed Algorithm:**
1. Start
2. Upload the dataset of code from Google Code Jam(GCJ)
3. Software piracy analysis:
   a. Preprocessing
   b. Tokenization
   c. Token’s weighting
   d. Normalization
4. Detect software piracy
   a. Apply Tensorflow neural network
   b. Detect piracy using similar source code
5. Return result/ Show result
6. End.

**III. EXPERIMENTAL RESULT:**
The trained dataset is uploaded from Google code jam (GCJ) for detecting software piracy. This data is preprocessed Tokenized weighted noisy data and extracting meaningful tokenise
The tokens are weighted using TF-IDF(Term frequency-Inverse document frequency).
The detection module applied the tensorflow neural network in order to detect software piracy using source code plagiarism and shows the result that how much the software is pirated in percentage.

In above screen we are uploading one B.cpp file and now click open to detect similarity or plagiarism score. It shows the number of train file loaded, total words in all file loaded and total train data size.
In above screen uploaded program contains no plagiarism and its score is 34%. If score > 50 % then it will consider as pirated.

In above screen we are uploading another program and below is the result

In above screen we can see for last uploaded program similarity score is 0.77% and its consider as pirated and its contains copying from which user that will also be displayed. In above screen sigsegv is the username in train folder from which this test file copied.
Now click on ‘Accuracy Graph’ button to get accuracy
In above screen x-axis represents propose and existing technique name and y-axis represents accuracy.

IV. CONCLUSION

The industrial IoT based network is rapidly growing in the coming future. The detection of software piracy are the main challenges in the field of cyber security using IoT-based big data. In this system proposed a combined deep learning based approach for the identification of pirated and malware files. First, the Tensor Flow neural network is proposed to detect the pirated feature of original software using software plagiarism. We collected 100 programmers’ source codes files from GCJ to investigate the proposed approach. The source code is prepossessed to clean from noise and to capture further the high-quality features which include useful tokens. Then, TFIDF and LogTF weighting techniques are used to zoom the contribution of each token in terms of source code similarity. The weighting values are then used as input to the designed deep learning approach. Secondly, we proposed a novel methodology based on convolution neural network and color image visualization to detect malware using IoT. We have converted the malware files into color images to get better malware visualized features. Then, system passed these visualized features of malware into deep convolution neural network. The experimental results show that the combined approach retrieve maximum classification results as compared to the state of the art techniques.


5. Donato Malerba, “Mining Spetial Data: Opportunities and challenges of Relation Approaches”,University of degli study, Italy.


10. Shahzad Qaiser, Ramsha Ali, “Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents”, School of Computer University, School of Quantitive Sciences University, Uttar Malaysia, July 2018.


