**IMPROVED DETECTION SYSTEM FOR ONLINE SOFTWARE PIRACY USING AN HYBRIDIZED ALGORITHM**

**TABLE OF CONTENT**

**CHAPTER ONE: INTRODUCTION**

* 1. **Background to the Study**

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 detect pirated software. The Proposed system will help to avoid the reputational 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 model will try to detect software piracy by providing less computational cost to improve the accuracy.

* 1. Statement of the Problem
  2. Aims and Objectives

1. To provide detection about malware and threads in files across network.
2. To detect software piracy using deep learning approach.
3. To identify pirated software using Tensor-Flow neural network.
4. To detect malicious infections using convolutional neural network.
5. To avoid economic and reputation damages causes due to threads.

1.4 Significance of the Study

* 1. Scope of the Study
  2. **Definition of Terms**

**Entropy:** It is a measure of randomness or unpredictability in the data set.

**Information Gain:** A measure of the decrease in the entropy after the data set is split is the information gain.

**Leaf Node:** A leaf node is a node that carries the classification or the decision.

**Decision Node:** A node that has two or more branches.

**Root Node:** The root node is the topmost decision node, which is where you have all of your data.

**CHAPTER TWO: LITERATURE REVIEW**

**2.1 Software Piracy**

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 re-construct or re-design the logic into other programming language or may change minor details of the software. It is very exasperating to catch such assaulter’s 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 reputation and economic damages. Nowadays every other software is pirated, there are many scenarios in which it can occur, and 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. In this project we proposed a combined machine learning algorithm approach to detect the pirated software.

**2.2 Types of Software Piracy**

It seems that illegal software is available anywhere, to anyone, at any time. The following are some of the methods by which illegal copies of software circulate among computer users.

**SOFTLIFTING**

The most common type of piracy, softlifting, (also called softloading), means sharing a program with someone who is not authorized by the license agreement to use it. A common form of softlifting involves purchasing a single licensed copy of software and then loading the software onto several computers, in violation of licensing terms. On college campuses, it is rare to find a software program that has *not* been softloaded. People regularly lend programs to their roommates and friends, either not realizing it is wrong, or not thinking that it's a big deal. Softlifting is common in both businesses and homes.

**HARD DISK LOADING**

Often committed by hardware dealers, this form of piracy involves loading an unauthorized copy of software onto a computer being sold to the end user. This makes the deal more attractive to the buyer, at virtually no cost to the dealer. The dealer usually does not provide the buyer with manuals or the original CDs of the software. This is how operating systems, like Windows 95, are often pirated.

**RENTING**

Renting involves someone renting out a copy of software for temporary use, without the permission of the copyright holder. The practice, similar to that of renting a video from Blockbuster, violates the license agreement of software.

**OEM UNBUNDLING**

Often just called "unbundling," this form of piracy means selling stand-alone software originally meant to be included with a specific accompanying product. An example of this form of piracy is someone providing drivers to a specific printer without authorization.

**COUNTERFEITING**

Counterfeiting means producing fake copies of a software, making it look authentic. This involves providing the box, CDs, and manuals, all designed to look as much like the original product as possible. Microsoft products are the ones most commonly counterfeited, because of their widespread use. Most commonly, a copy of a CD is made with a CD-burner, and a photocopy of the manual is made. Counterfeit software is sold on street corners, and sometimes unknowingly sold even in retail stores. Counterfeit software is sold at prices far below the actual retail price.

**ONLINE PIRACY**

The fastest-growing form of piracy is Internet piracy. With the growing number of users online, and with the rapidly increasing connection speeds, the exchange of software on the Internet has attracted an extensive following. In the past, bulletin board systems (BBS) were the only place where one could download pirated software. Currently, there are hundreds of thousands of "warez" sites providing unlimited downloads to any user. Often, the software provided through these "warez" sites is cracked to eliminate any copy protection schemes.

2.3Reasons for Using Only Genuine Software

2.4 Features of Genuine Software

2.5 Existing Preventive Measures for Software Piracy

2.6 Factors behind Software Piracy

2.7 Software Piracy in Developed and Developing Countries

2.7 Risk involved in Using Pirated software

**2.8 Random Forest Algorithm**

A Random Forest Algorithm is a supervised machine learning algorithm which is extremely popular and is used for Classification and Regression problems in Machine Learning. We know that a forest comprises numerous trees, and the more trees more it will be robust. Similarly, the greater the number of trees in a Random Forest Algorithm, the higher its accuracy and problem-solving ability.  Random Forest is a classifier that contains several decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset. It is based on the concept of ensemble learning which is a process of combining multiple classifiers to solve a complex problem and improve the performance of the model.

**Types of Machine Learning**

To better understand Random Forest algorithm and how it works, it's helpful to review the three main types of [machine learning](https://www.simplilearn.com/tutorials/machine-learning-tutorial/what-is-machine-learning);

1. **Reinforced Learning**

The process of teaching a machine to make specific decisions using trial and error.

1. **Unsupervised Learning**

Users have to look at the data and then divide it based on its own algorithms without having any training. There is no target or outcome variable to predict nor estimate.

1. **Supervised Learning**

Users have a lot of data and can train your models. Supervised learning further falls into two groups: classification and regression.

With supervised training, the training data contains the input and target values. The [algorithm](https://www.simplilearn.com/10-algorithms-machine-learning-engineers-need-to-know-article) picks up a pattern that maps the input values to the output and uses this pattern to predict values in the future. Unsupervised learning, on the other hand, uses training data that does not contain the output values. The algorithm figures out the desired output over multiple iterations of training. Finally, we have reinforcement learning. Here, the algorithm is rewarded for every right decision made, and using this as feedback, and the algorithm can build stronger strategies.

# 2.8.1 How Random Forest algorithm works

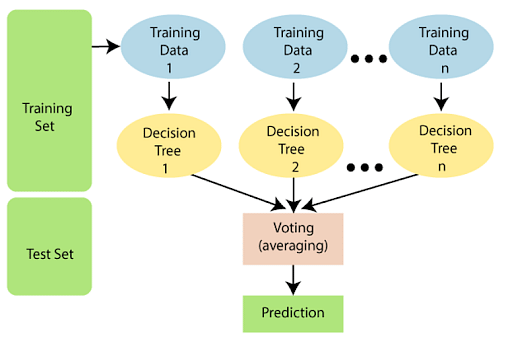
The following steps explain the working Random Forest Algorithm:

Step 1: Select random samples from a given data or training set.

Step 2: This algorithm will construct a decision tree for every training data.

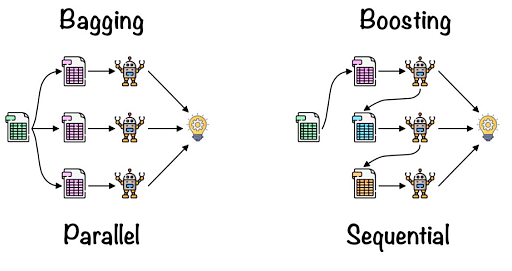
Step 3: Voting will take place by averaging the decision tree.

Step 4: Finally, select the most voted prediction result as the final prediction result.

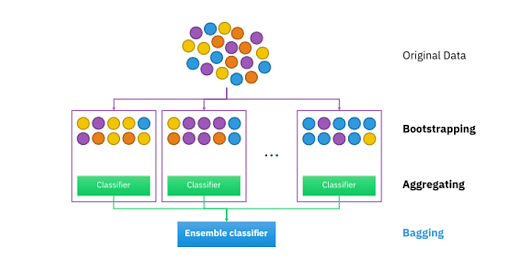


This combination of multiple models is called Ensemble. Ensemble uses two methods:

1. **Bagging:** Creating a different training subset from sample training data with replacement is called Bagging. The final output is based on majority voting.
2. **Boosting:** Combing weak learners into strong learners by creating sequential models such that the final model has the highest accuracy is called Boosting. Example: ADA BOOST, XG BOOST.



**Bagging**: From the principle mentioned above, we can understand Random forest uses the Bagging code. Now, let us understand this concept in detail. Bagging is also known as Bootstrap Aggregation used by random forest. The process begins with any original random data. After arranging, it is organised into samples known as Bootstrap Sample. This process is known as Bootstrapping. Further, the models are trained individually, yielding different results known as Aggregation. In the last step, all the results are combined, and the generated output is based on majority voting. This step is known as Bagging and is done using an Ensemble Classifier.



# 2.8.2 Advantages of Random Forest Algorithm

1. Can perform both Regression and classification tasks.
2. Produces good predictions that can be understood easily.
3. Can handle large data sets efficiently.
4. Provides a higher level of accuracy in predicting outcomes over the decision algorithm.

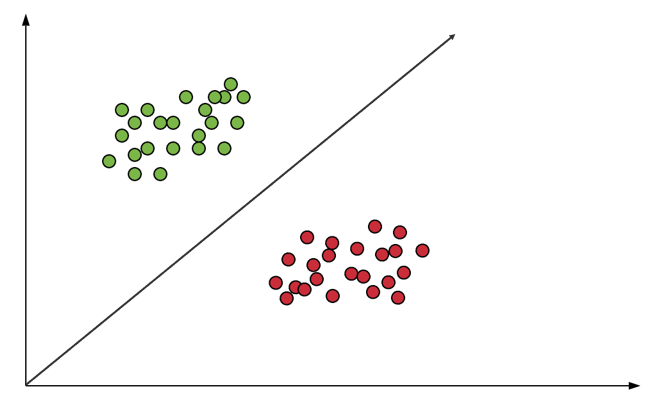
**2.9 Support Vector Machine Algorithm**

A Support Vector Machine was first introduced in the 1960s and later improvised in the 1990s. It is a[supervised learning](https://www.edureka.co/blog/supervised-learning/) machine learning classification algorithm that has become **extremely popular** nowadays owing to its extremely efficient results. An SVM is implemented in a slightly different way than other machine learning algorithms. It is capable of performing classification, [regression](https://www.edureka.co/blog/linear-regression-in-python/) and outlier detection.

Support Vector Machine is a discriminative classifier that is formally designed by a separative hyperplane. It is a representation of examples as points in space that are mapped so that the points of different categories are separated by a gap as wide as possible. In addition to this, an SVM can also perform non-linear classification.

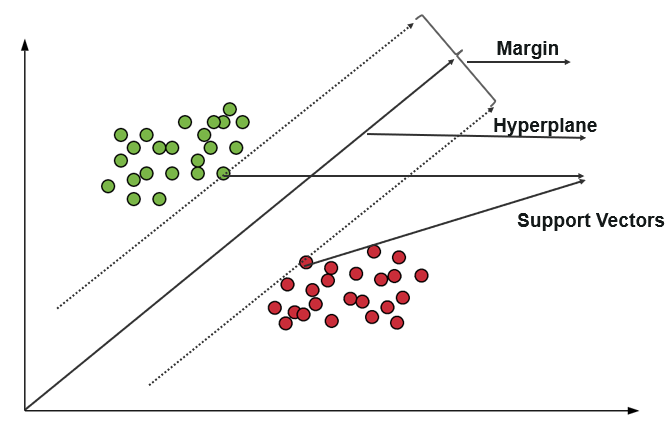
# 2.9.1 How Support Vector Machine Algorithm works

The main objective of a support vector machine is to segregate the given data in the best possible way. When the segregation is done, the distance between the nearest points is known as the margin. The approach is to select a hyperplane with the maximum possible margin between the support vectors in the given [data-sets](https://www.edureka.co/blog/25-best-free-datasets-machine-learning/).



To select the maximum hyperplane in the given sets, the support vector machine follows the following sets:

* Generate hyperplanes which segregates the classes in the best possible way
* Select the right hyperplane with the maximum segregation from either nearest data points

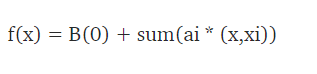
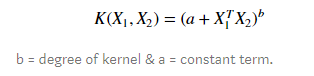
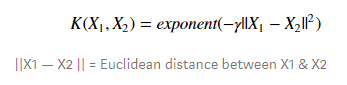


**How to deal with inseparable and non-linear planes**

In some cases, hyperplanes cannot be very efficient. In those cases, the support vector machine uses a kernel trick to transform the input into a higher-dimensional space. With this, it becomes easier to segregate the points. Let us take a look at the SVM kernels.

**SVM Kernels**

An SVM kernel basically adds more dimensions to a low dimensional space to make it easier to segregate the data. It converts the inseparable problem to separable problems by adding more dimensions using the kernel trick. A support vector machine is implemented in practice by a kernel. The kernel trick helps to make a more accurate classifier. Let us take a look at the different kernels in the Support vector machine.

* **Linear Kernel** – A linear kernel can be used as a normal dot product between any two given observations. The product between the two vectors is the sum of the multiplication of each pair of input values. Following is the linear kernel equation.  
  
* **Polynomial Kernel** – It is a rather generalized form of the linear kernel. It can distinguish curved or nonlinear input space. Following is the polynomial kernel equation.  
  
* **Radial Basis Function Kernel** – The radial basis function kernel is commonly used in SVM classification, it can map the space in infinite dimensions. Following is the RBF kernel equation.  
  

# 2.9.2 Advantages of Support Vector Machine Algorithm

# Effective in high dimensional spaces

1. Still effective in cases where the number of dimensions is greater than the number of samples
2. Uses a subset of training points in the decision function that makes it memory efficient
3. Different kernel functions can be specified for the decision function that also makes it versatile

**2.10 Navies Bayes Algorithm**

The naive Bayes Algorithm is one of the popular classification machine learning algorithms that helps to classify the data based upon the conditional probability values computation. It implements the Bayes theorem for the computation and used class levels represented as feature values or vectors of predictors for classification. Naive Bayes Algorithm is a fast algorithm for classification problems. This algorithm is a good fit for real-time prediction, multi-class prediction, recommendation system, text classification, and sentiment analysis use cases. Naive Bayes Algorithm can be built using Gaussian, Multinomial and Bernoulli distribution. This algorithm is scalable and easy to implement for a large data set.

It helps to calculate the posterior probability using the prior probability of class the prior probability of predictor P(x), and the probability of predictor given class, also called as likelihood

The formula or equation to calculate posterior probability is:

# 2.10.1 How Navies Bayes algorithm works

Let us understand the working of the Naive Bayes Algorithm using an example. We assume a training data set.

**Step 1:**Make Frequency Tables Using Data Sets.

**Step 2:** Make a likelihood table by calculating the probabilities of each condition.

**Step 3:** Now, we need to calculate the posterior probability using the Naive Bayes equation for each class.

# 2.10.2 Advantages of Navies Bayes Algorithm

1. Easy to implement.
2. Fast
3. If the independence assumption holds, then it works more efficiently than other algorithms.
4. It requires less training data.
5. It is highly scalable.
6. It can make probabilistic predictions.
7. Can handle both continuous and discrete data.
8. Insensitive towards irrelevant features.
9. It can work easily with missing values.
10. Easy to update on the arrival of new data.
11. Best suited for text classification problems.

**2.11 Related Work**

According to Author, of Camerino University, there are various challenges in cyber security one of them include software piracy. Currently, every other available installed software application is pirated. Software piracy is the act of copying distributing, using software illegally. The cracker or attacker may crack the original software and re-structure it into his own form. As different programming languages have different semantic and syntactical structures, the crackers may redesign the software into some another type of programming language. In this reference, they proposed a combined deep learning approach to identify the pirated and malware attacks on industrial IOT cloud. The deep neural network is designed to capture the pirated software using source code plagiarism. Further, the deep convolutions neural network is designed to capture the malicious patterns of malware through binary visualization. The combined solutions of the proposed approach are much promising in terms of classification performance.

According to Basel Halak and Mohammad El-Hajjar, they adapted the plagiarism prevention and detection techniques. Plagiarism is stealing someone else’s work or publication and representing them as on original work. According to this reference of the paper there are mainly two techniques of preventing plagiarism. First is about plagiarism prevention using unique assignment and second is about plagiarism prevention using Individual presentation. In which we detect and prevent plagiarism based on unique specification this type of plagiarism is more difficult to highlight using common plagiarism detection tool. The use of individual presentation technique can also be effective in detecting plagiarism of undocumented ideas in group design project. In this reference, these two techniques is based on the use of individual design specification in the course works and individual presentation of the work. They gave examples where these techniques have been used in different modules in the University of Southampton and have shown how this techniques can be effective in reducing plagiarism and at the same time improving students understanding.

According to Dept. of computer software Hanyang University, Korea, they suggested a software plagiarism using API- labeled control flow graph (A-CFG). A- CFG represents the sequences and frequencies of API which are rarely changed by semantic preserving transformation attacks. They performed scalable comparison between A-CFG as showing each A-CFG as a single score vector through RwR. Ex: Results shows that our proposed system outperforms existing methods in terms of both accuracy and creditability in a reasonable computational time.

According to Author, Department of MCA, PES institute of technology proposed that estimate 60% of data mining tasks are mainly spend on preprocessing the data. So, they performed a cyclomatic complexity analysis which is used to indicate the complexity of program which is turn enables us to calculate the number of coding errors along with their source code complexity. Through this phase errors are separated both at instance and schema level from single and multiple source of data followed by this the second phase involves a sequential flow analysis for data preprocessing of spatial data, multidimensional data, web log data etc. It was observed that after performing this two phases, preprocessing is improved for further data mining process. It is known that data cleaning or data cleansing is one of the measure step in KDD process of data meaning two generate accurate and appropriated data for different data mining tasks. Data may come from various source such as homogenous databases, heterogeneous databases, flat files and other.

It is well known that over 80% of the time required to carry out any real world data mining project is usually spent on data preprocessing. Data preprocessing lays the groundwork for data mining. Web mining is to discover and extract useful information from the World Wide Web. It involves the automatic discovery of patterns from one or more web servers. This helps the organizations to determine the value of specific customers, cross marketing strategies, etc. A complete preprocessing techniques is being proposed to preprocess the web log for extraction of user patterns. Data cleaning algorithm removes the irrelevant entries from web log and filtering algorithm discards the uninterested attributes from log file. An important task in any data mining application is the creation of a suitable target data set to which data mining and statistical algorithm can be applied. This is particularly important in web usage mining due to the characteristics of click stream data and this relationship to other related data collected from multiple sources and across multiple channels. The data preparation process is often the most time consuming and computationally intensive step in the web usage mining process, and often requires the use of special algorithms and heuristics, not commonly employed in other domains.

**CHAPTER THREE: MATERIALS AND METHODS**

3.1 Analysis of the Existing System

3.2 Algorithm of the Existing System

3.3 Advantages of Existing System

3.4 Disadvantages of Existing system

3.5 Analysis of Proposed System

**3.6 Algorithm of the Proposed System**

The proposed system will be using the following algorithms, Random Forest Algorithm, Naïve Bayes and Support Vector Machine Algorithm.

3.6.1 Random Forest Algorithm and how it works in this research

3.6.2 Navies Bayes and how it works in this research

3.6.3 Support Vector Machine Algorithm and how it works in this research

3.7 Design of the Proposed System

3.8 Components of the Proposed System Architecture

3.9 Use Case Activity Diagram of the Proposed System Design

3.10 UML Diagram of the Proposed System.

3.11 Sequence Diagram of the Proposed System

3.12 High Level Diagram of the Proposed System

3.13 Methodology of the Proposed System

3.14 Justification of the Design Methodology Used

3.15 Datasets

The data set used in this project was collected from Kaggle.

3.16 Advantages of the Proposed System

3.17 Choice and Justification of Programming Language for the Proposed System

**CHAPTER FOUR: RESULTS AND DISCUSSION**

4.1 Implementation of the Proposed System

4.1.1 The Pre-processing Module

4.1.2 The Processing Module

4.1.3 The Post Processing Module

4.2 System Requirement of the Proposed System

4.2.1 Hardware Requirements

4.2.2 Software Requirements

4.3 Documentation

4.4 Results and Discussions

4.4.1 Genuine Software Detection Results and Discussion

4.4.2 Pirated Software Detection Results and Discussion

4.4.3 Malware Detection Results and Detection

4.5 Results and Performance Evaluation of Proposed System

4.5.1 Comparison of the Results and Performance Evaluation of Proposed System and Existing System

4.5 Accuracy Calculation

**CHAPTER FIVE: SUMMARY, CONCLUSION AND RECOMMENDATION**

5.1 Summary of Finding

5.2 Conclusion

5.3 Recommendation

5.4 Contribution to Knowledge

**REFERENCES**

[1] Sohail Jabar, Kaleem R. Malik, Mudassar Ahmad, Omar Aldabbas, Muhammad Asif, Shehzad Khalid, Kijun Han,”A Methodology of Real-Time Data Fusion for Localized Big Data Analytics”,March 15, 2018.

[2] Manisha Mishra, Monika Srivastava, “A view of Artifical Neural Network”,Dr. Virendra Swarup Group of Institution Unnao,2014.

[3] Liping Yuan, Zhiyi Qu, Yufong Zhao, Hongshuai Zhang, Qing nian, “A convolutional neural network based on TensorFlow for face recognition “Lanzhou University,China,2017

[4] Farhan Ullah, Hamad Naeem, Sohail Jabbar, Shehzad Khalid, Muhammad Ahsan, Latif Fadi AL-turjman and Leonardo Mostarda, “cyber security threads detection in internet of things using deep learning approach,” Sichuan University, Chengdu 610065,China,2017.

[5] Donato Malerba, “Mining Spetial Data: Opportunities and challenges of Relation Approaches ”,University of degli study, [1] Italy.

[6] Basel Halak, Mohammed El-Hajjar, “Plagiarism Detection and Prevention Techniques In Engineering Education”, University of Southampton , Southampton, UK, 2016.

[7] Dong-kyu, Jiwoon Ha, Sang-wook kim, BooJong Kang “A Software plagiarism detection : A graph-based approcach”, Hanyang University, October 2013.

[8] P.Sreenivas, Dr.C.V.Srikrishna, “An Analytical approach for Data Preprocessing”, PES Institute of Technology, 27 February 2014.

[9] Faith Ertam, Galip Aydin, “Data Classification with Deep Learning using Tensorflow”, Firat University, Elazig, Turkey, 2017.

[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.

[11] Prafulla Bafna, Dhanya Pramod, Anagha Vaidya, “Document Clustering: TF-IDF Aprroach”,Symbiosis International University, pune,2016.

[12] Vijayashri Losarwar, Dr. Madhuri Josji, “Data Preprocessing in Web Usage Mining”,Singapore, July 15-16, 2012.

[13] Jin Guo, ”Critical Tokenization and its properties ”, National University of Singapore. [14] Fco.Mario Barcala, Jesus Vilares, Miguel A. Alonso. Jorge Grana, Manuel Vilares“Tokenization and Proper Noun Recognition for Information Retrival” Departmento de Computacion, Universidade da Coruna Campus de Elvina s/n, 15071 La Coruna, Spin,2002.

[15] Dong-Kyu Chae, Jiwoon Ha, Sang –Wook Kim, BooJoong Kang, Eul Gyu Im, “Software Plagiarism Detection: A Graph-Based Approach”, Hanyang University, Korea, 19 June 2015.

[16] Elfwing, S., E. Uchibe,K.Doya, “Sigmoid-weighted linear units for neural network function approximation in reinforcement learning”, Neural Networks, 2018 .

[17] Ullah,F, “Software plagiarism detection in multi-programming languages using machine learning approach. Concurrency and Computation: Practice and Experience”2018.

**APPENDICES**

APPENDIX A: Source Codes

APPENDIX B: Results