Applied Machine Learning to Detection of Steganography.

Jamie Hoang (jah1018@my.londonmet.ac.uk), ID: 19009101
BSc. Digital Forensic and Cyber Security

Supervisor: Mohamed Chahine Ghanem

London Metropolitan University - [13/10/2021]

Introduction

The following project proposed is an attempt to incorporate machine learning into the field of Digital Forensics. Forensic Investigators may face certain time crunches during the course of a case from upcoming deadlines or vast workloads. Applying machine learning within this field aims to reduce the pressure faced by Forensic Investigators. Within this project, the student will attempt to create an application to detect steganography within files. Steganography can be considered a manually demanding workload for Forensic Investigators as Criminals hide sensitive data within innocuous files. In a real world application, this software would flag suspicious files for manual review by a human thus reducing the workload for investigators of having to search an entire storage medium.

Ideally, this application would learn from real-world data made by actual criminal enterprises. Therefore the algorithm would learn from essentially the best that the world has to offer. Instead, we will be using samples created by the student himself however using a range of steganography tools and methodologies.

Aims and Objectives

The primary goal of this project is to create a piece of software that utilizes machine learning to detect steganography use within images. Which will be achieved with the objectives listed below.

- To assemble and gather the necessary knowledge to build said application.
    - Re-learning the basics of Python and principles of using a programming language.
    - Learning the fundamentals of machine learning and its mathematical algorithms. To grasp how the application should operate on a principle level.
    - Compiling the necessary Python Libraries and their relevant documentation.

- To document findings and progress in a timely and concise manner.
    - Attending periodical meetings and documentation of all aspects of the project. Such as explaining the theoretical side, explanations of the software and personal reflections from the student.

- To create training data for the machine learning algorithm to learn from. Creating a sample size that is

diverse and challenging to the algorithm.
    - This would mean compiling various software and methodologies in order to apply Steganography to files.
    - The sample data should be in an Excel Spreadsheet for outsiders to review the sample data. With additional metadata on how each individual sample was created.

- For the machine learning algorithm to achieve a 95% accuracy rate when assessing files.
    - In the real world, the minimum accuracy rate should be 99%.however due to resource constraints. A 5% error rate is acceptable.
    - Tests to quantify an accuracy rate can be done with internal mechanisms and external tests done with samples separate from the training data.

This is further broken down into simple tasks that can be plotted on a gantt chart.

- Literature Search
- Literature Review
- Design Software
- Develop and Iterate Versions
- Collect and process samples
- Document Samples
- Train software
- Software Evaluation
- Final Report

Expected outcomes and Deliverables

- A software application incorporating machine learning and necessary python libraries. In which it can analyze files to flag suspicious files for suspected steganography use.
    - With additional documentation, explaining its internal processes and a breakdown of the code structure.

- Test results showcasing the software application achieving or failing to achieve the expected 95% accuracy rate.

- Concise spreadsheet showcasing all created steganography samples with relevant metadata on its creation.

- A final report summarizing the progress, findings and outcome of this project.

Methodology

As Stated previously, the aims and objectives of the project were broken down into sequential tasks.

- Literature Search
- Literature Review
- Design Software
- Develop and Iterate Versions
- Collect and process samples
- Document Samples

- Train software
- Software Evaluation
- Final Report

As the following list above it is ordered from the first task to the last to be completed sequentially. This section of the document will go into greater detail how these specific tasks are accomplished.

Literature Search, Literature Review and Design Software

In the early stages of the project. A considerable amount of time will be spent on research to ensure that the student has necessary skills and knowledge to implement the core functions of the product. This will mostly consist of searching for relevant documentation, toolsets, software and learning materials. After which we can start the software development process. This is due to the student's inexperience with machine learning and programming.

This will consist primarily of secondary research. As most of the resources can simply be obtained from the internet and from the university. This is due to the resource constraints of the student. This can be done by reading into the various libraries constructed around programming languages. These libraries can be viewed as what enables functions within software. For example, a library can be dedicated to data processing which enables software to process data. Through reading various library documentations we can identify which libraries will be needed for this project.

Then the design of the software can be based around which libraries were picked. The process for designing the software will be kept to a singular week as most of the pre-planning and design work can be informed by relevant library documentation. As they can provide useful tips and examples for developers. In addition, other resources could further aid in understanding and clarity in chosen software libraries if it's relevant documentation isn't sufficient.

Develop and Iterate Versions

We will be using the following software development methodology "SCRUM". Which allows us to adapt depending on if we are on schedule, behind schedule or exceeding expectations. It is considered to be an iterative methodology that builds upon what happened prior. It breaks down the needs of the product into a backlog. Then it is further broken down into sprints. Which are short bursts of time in which a planned action is set as a target. Such as the implementation of a core function within a piece of software. These sprints are then repeated until there isn't a product backlog anymore. Thus resulting in the final product. Whilst initially meant for small teams. The student feels that it can be applied to a single individual.

A visual illustration of this can be seen in Figure 4.a below.

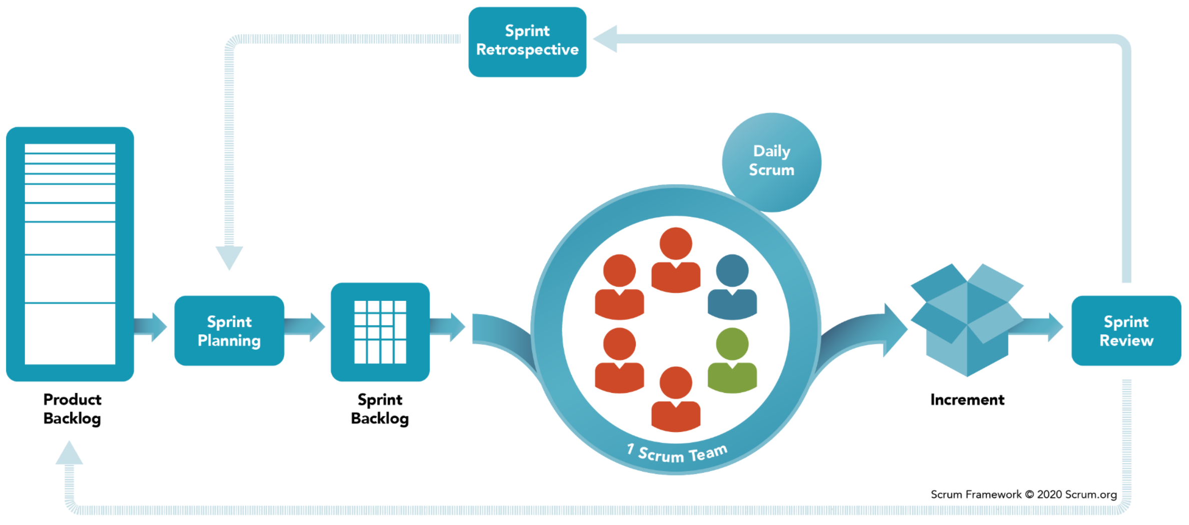


Figure 4.a - SCRUM Diagram (Scrum.org, 2021)

The core functions will be first implemented. Which will consist of creating inputs for the software, creating a system for training the software with selected data, inspect files and output results. Then any additional non-essential features can be implemented if there is any spare development time.

Collect and process samples, Document Samples

The process of creating steganography samples will be determined from research during the literature review task. The tools selected will then be subjected to create over a thousand samples. As well as, three different and distinctive steganography tools will be used. The size and variety of samples is key to training software to detect steganography as a harsh difficulty needs to be given to said software. Through challenge, it would make key comparisons between samples and develop an algorithm to achieve the set goals.

The samples would be stored in various locations. Namely a physical and on a cloud service. In addition, an excel spreadsheet would be created to document relevant metadata on said samples. Detailing its date of creation, what steganography tool was used to create said sample and the sample itself. In addition, the samples would be processed into a format that the software can recognize and learn from.

Train Software and Software Evaluation

As data is fed into the inputs of the software. It would be merely an automated process with minor oversight from the student. As software itself attempts to establish links between what makes a suspect for steganography use and what isn't. The overall pool of samples would be split into two. Training set and Testing set. The training set being the samples initially fed into the software so it would learn and establish links. The testing set is separate from the training set as it is purely used to gauge the accuracy rate of the software. The end result being the software evaluation where we would test the software with the testing set and attain a tangible accuracy rate.

Resource Requirements

- A collection of Anti-Forensic software intended to be used for creation of steganography samples.
    - Examples of such software are “Hermetic Stego”, “Slacker”, “S-Tools” and “Camouflage”

- Hardware / Cloud Based Platforms to facilitate the training of the application via Machine Learning.
    - Google’s “Colaboratory” is one of the considered cloud-based platforms. As it is free for use by students.

- Documentation or learning resources on the principles of Machine Learning.

- Relevant python documentation and python libraries needed to achieve the aims and objectives of the project.

- (Optional) Steganography samples / datasets created by criminal enterprises that could be available to the public intended for training purposes.

Project Plan

Within this section of the document, It is visually illustrated how the tasks are structured and projected time for each specified tasks. As seen in Figure 6.a below, the project is broken down into a Work Breakdown Structure.

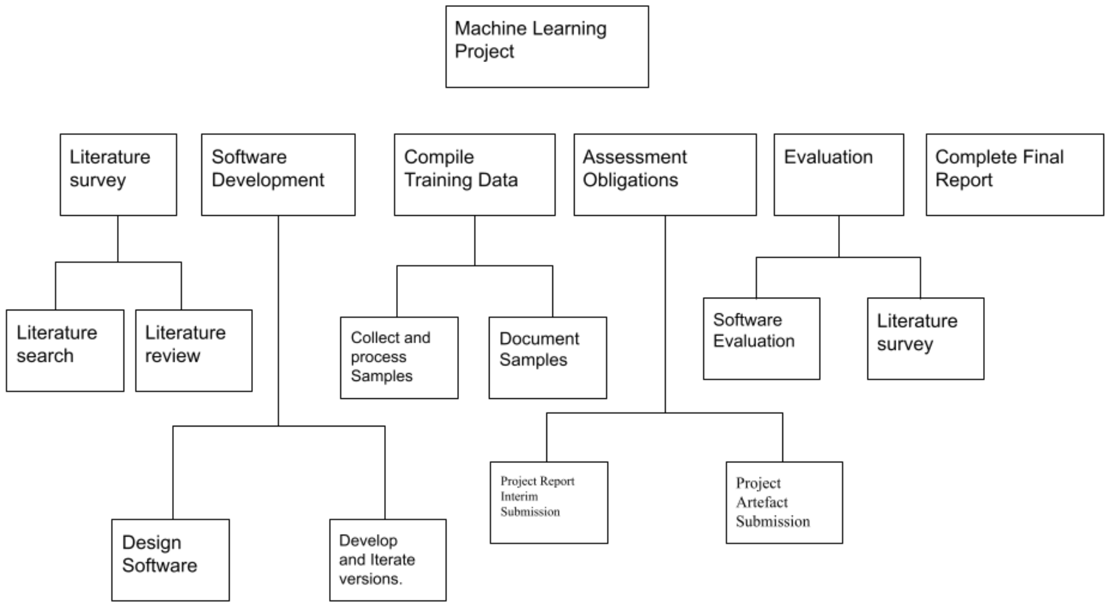


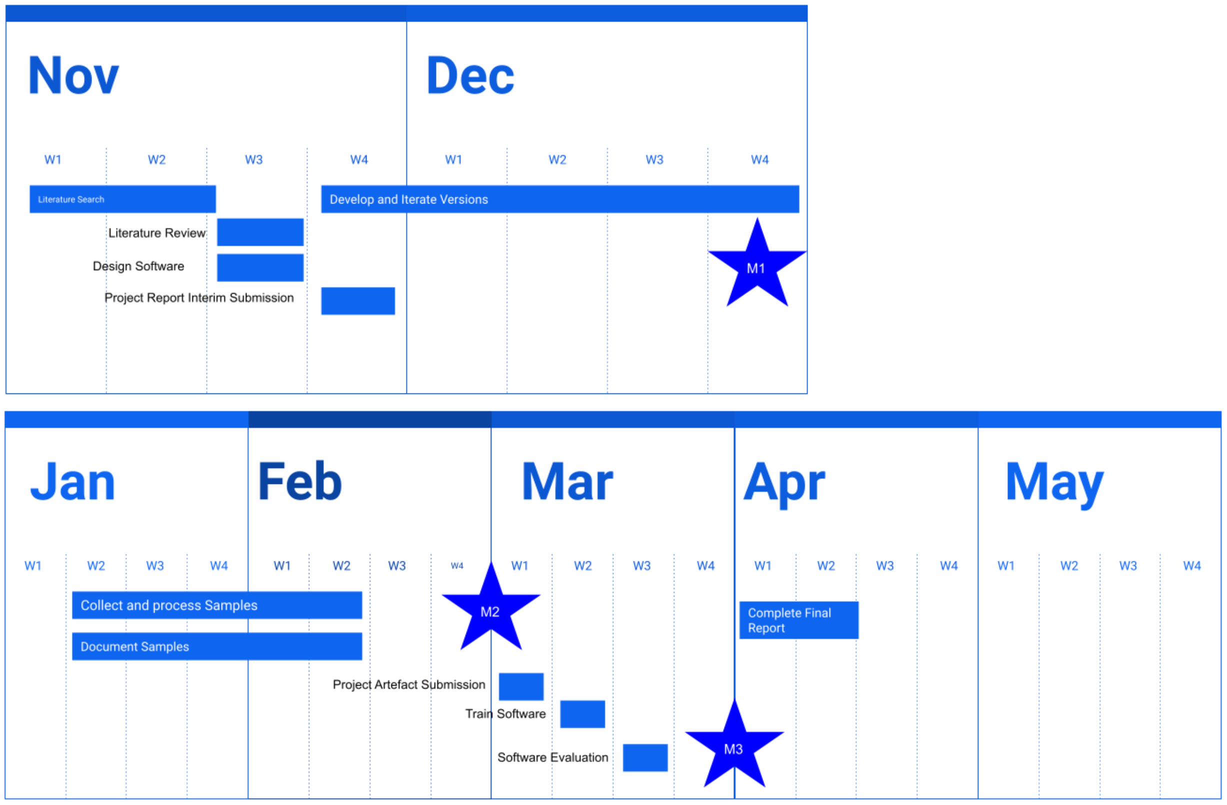
Figure 6.a - Work Breakdown Structure

Tasks were created in accordance with the SMART Technique. Which is an acronym for Specific, Measurable, Achievable. Realistic and Timely. As all of the tasks listed follow these qualities. Which demonstrates the feasibility of this project. The projected duration of each task is listed below. Within Figure 6.b.

No.	Activities	Estimate Duration	Activity Description
1	Literature Search	2 Weeks	Conducting research and reading into various resources, compiling needed knowledge and skillsets through whatever means necessary.
2	Literature Review	1 Week	Reviewing said research, drawing conclusions and formulating theories or conclusions.
3	Design Software	1 Week	Planning the structure of the software, as well as its core functions.
4	Develop and Iterate Versions	5 Weeks	Development phase of the software. Where the software is actually created and improved upon in each installment.
5	Collect and process Samples	5 Weeks	The creation of steganography samples utilizing various tools.
6	Document Samples	5 Weeks	Recording and assigning tags to various samples so that they can be independently reviewed.
7	Project Report Interim Submission	1 week	As part of the Assignment obligations, it is the first report after the project proposal.
8	Project Artefact Submission	1 week	As part of the Assignment obligations, it is the second report after the project proposal.
9	Train Software	1 Week	The feeding of samples into the software, so that it can further develop and establish links with the data provided.
10	Software Evaluation	1 Week	The testing of the software to see if the project goals were met.
11	Complete Final Report	2 week	The documentation that summarizes the findings and results of this project.
	Total Duration	25 Weeks	

Figure 6.b - Task Listing Table

Lastly, the tasks are plotted out on a gantt chart. Which illustrates when tasks will be completed during the course of the year. Some tasks can be completed in parallel with other tasks. In addition, milestones are listed in which a batch of tasks must be completed before the proposed deadline date. This ensures that the project is on track and on schedule.



Milestone No.	
1	31st of December of 2021
2	28th of February of 2022
3	31st of March of 2022

Figure 6.C - Milestone Table and Gantt Chart

Bibliography & References

TensorFlow.org (2021) API Documentation. Available at: https://www.tensorflow.org/api_docs (Accessed: 13/10/2021)

Python Software Foundation (2021) Python 3.10.0 documentation. Available at: https://docs.python.org/3.10/ (Accessed: 13/10/2021)

the pandas development team (2021) pandas documentation. Available at: https://pandas.pydata.org/docs/ (Accessed: 13/10/2021)

Arthur L. Samuel (1959) Some studies in machine learning using the game of checkers. Available at: https://hci.iwr.uni-heidelberg.de/system/files/private/downloads/636026949/report_frank_gabel.pdf

IBM (2021) What is Machine Learning? Available at: https://www.ibm.com/uk-en/cloud/learn/machine-learning (Accessed: 13/10/2021)

scikit-learn developers (2021) User Guide. Available at: https://scikit-learn.org/stable/user_guide.html (Accessed: 13/10/2021)

Savita Pahuja (2015) Scrum for Individuals. Available at: https://www.infoq.com/news/2015/02/personal-scrum/ (Accessed: 13/10/2021)

Scrum.org (2021) The Scrum Framework. Available at: https://www.scrum.org/resources/what-is-scrum (Accessed: 20/10/2021)