A CRYPTOJACKING DETECTION SYSTEM WITH PRODUCT MOMENT CORRELATION COEFFICIENT (PMCC) HEATMAP INTELLIGENT

KONG JUN HAO

Bachelor of Computer Science (Software Engineering) with Honours

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

I declare that this thesis is classified as:

I acknowledge that Universiti Malaysia Pahang reserves the following rights:

1. The Thesis is the Property of Universiti Malaysia Pahang
2. The Library of Universiti Malaysia Pahang has the right to make copies of the thesis for the purpose of research only.
3. The Library has the right to make copies of the thesisfor academic exchange.

Certified by:

(Student's Signature)	(Supervisor's Signature)
Name: KONG JUN HAO Date: 3rd February 2023	Name: TS.DR.AHMAD FIRDAUS BIN ZAINAL ABIDIN Date: 3rd February 2023

NOTE : * If the thesis is CONFIDENTIAL or RESTRICTED, please attach a thesis declaration letter.

THESIS DECLARATION LETTER (OPTIONAL)

Librarian,
Perpustakaan Universiti Malaysia Pahang,
Universiti Malaysia Pahang, 26600,
Pekan, Pahang, Malaysia.

Dear Sir,

CLASSIFICATION OF THESIS AS RESTRICTED

Please be informed that the following thesis is classified as RESTRICTED for a period of three (3) years from the date of this letter. The reasons for this classification are as listed below.
Author's Name

Thesis Title

Reasons (i)

(ii)

(iii)

Thank you.

Yours faithfully,

(Supervisor's Signature)

Date: 13 Feb

Stamp: DR. AHMAD FIRDAUS BIN ZAINAL ABIDIN
SENIOR LECTURER
FACULTY OF COMPUTING
COLLEGE OF COMPUTING & APPLIED SCIENCES
UNIVERSITI MALAYSIA PAHANG
26600 PEKAN, PAHANG DARUL MAKMUR
TEL : 09-424 4629 FAX : 09-424 4666

Note: This letter should be written by the supervisor, addressed to the Librarian, *Perpustakaan Universiti Malaysia Pahang* with its copy attached to the thesis.

SUPERVISOR's DECLARATION

I/We* hereby declare that I/We* have checked this thesis/project* and in my/our* opinion, this thesis/project* is adequate in terms of scope and quality for the award of the degree of *Doctor of Philosophy/ Master of Engineering/ Master of Science in ……..

(Supervisor's Signature)

Full Name : AHMAD FIRDAUS BIN ZAINAL ABIDIN

Position : Senior

Date : 13 Feb

(Co-supervisor's Signature) Full Name :

Position :

Date :

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at Universiti Malaysia Pahang or any other institutions.

J.H.Kong

(Student's Signature)

Full Name : KONG JUN HAO

ID Number : CB19109

Date : 22/1/2023

A CRYPTOJACKING DETECTION SYSTEM WITH PRODUCT MOMENT CORRELATION COEFFICIENT (PMCC) HEATMAP INTELLIGENT

KONG JUN HAO

Thesis submitted in fulfilment of the requirements for
the award of the degree of
Bachelor of Computer Science (Software Engineering) with Honours

Faculty of Computing

UNIVERSITI MALAYSIA PAHANG

JANUARY 2023

ACKNOWLEDGEMENTS

I'd want to express my gratitude to my supervisor, Dr. Ahmad Firdaus Bin Zainal Abidin, who has been assisting me throughout my final year project from beginning to end. He had been assisting me with numerous issues as well as correcting my errors, particularly in the report. This endeavour would not be possible without his direction. Aside from that, I'd like to express my gratitude to my family for their unwavering support during the endeavour. They provided me with invaluable moral support in completing this project. A special appreciation is also extended to those who assisted me when I was having difficulty with this project. Their assistance is critical in helping me develop my system.

ABSTRACT

Cryptocurrency, often known as electronic money, is a currency that exists in digital form. As a result, numerous attackers or hackers are taking advantage of this chance to employ cryptojacking to gain access to a victim's computer or other device resources and mine cryptocurrency without the users' permission. Because the number of cryptojacking attacks is on the rise, this project use machine learning to detect cryptojacking. However, the feature of the data is too many, lowering the machine-learning detection prediction. Hence, a feature selection method is necessary to pick the right features. Aside from that, the objective of this project is to investigate cryptojacking in cryptocurrency users' devices, develop a machine learning model to detect cryptojacking, and evaluate the machine learning model's accuracy, true positive rate (TPR), false positive rate (FPR), and precision in detecting cryptojacking. This project research will present the PMCC Heatmap to choose the optimal attributes for using machine learning to detect cryptojacking in order to utilise machine learning to detect embedded malware. Furthermore, a random forest model is used in this study's machine learning classification. At the end of the process, the system will utilise this model to detect cryptojacking and users will be able to detect new cryptojacking malware based on the model. This study aims to develop a cryptojacking detection system based to the random forest algorithm.

ABSTRAK

Cryptocurrency, yang sering dikenali sebagai wang elektronik, adalah mata wang yang wujud dalam bentuk digital. Kesannya, banyak penyerang atau penggodam siber mengambil kesempatan peluang ini untuk menggunakan cryptojacking untuk mendapatkan akses kepada komputer mangsa atau sumber peranti lain dan melombong cryptocurrency tanpa kebenaran pengguna. Oleh demikian, jumlah serangan cryptojacking semakin meningkat, projek ini menggunakan pembelajaran mesin untuk mengesan cryptojacking. Walau bagaimanapun, ciri data yang terlalu banyak akan menurunkan ketepatan pembelajaran mesin dalam mengesan cryptojacking. Oleh itu, kaedah pemilihan ciri diperlukan untuk menapis dan memilih ciri yang paling tepat. Selain itu, objektif projek ini adalah untuk menyiasat cryptojacking dalam peranti pengguna cryptocurrency dan membangunkan model pembelajaran mesin untuk mengesan cryptojacking, berserta menilai ketepatan model pembelajaran mesin. Kadar positif sebenar (TPR), kadar positif palsu (FPR), dan ketepatan dalam mengesan cryptojacking akan dinilai untuk mencapai objektif projek ini. Penyelidikan projek ini akan membentangkan penggunaan PMCC Heatmap untuk memilih atribut optimum berserta penggunaan pembelajaran mesin dalam proses mengesan perisian hasad cryptojacking yang tertanam dalam peranti komputer pengguna. Tambahan pula, model hutan rawak dipilih sebagai klasifikasi pembelajaran mesin dalam pengajian projek ini. Pada akhir proses, sistem akan menggunakan model ini untuk mengesan cryptojacking dan pengguna akan dapat mengesan perisian hasad cryptojacking yang tersembunyi dalam peranti berdasarkan model ini. Kajian ini bertujuan untuk membangunkan sistem pengesanan cryptojacking berdasarkan algoritma hutan rawak.

TABLE OF CONTENT

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES

LIST OF APPENDIXES

LIST OF ABBREVIATIONS

TPR	True Positive Rate
FPR	False Positive Rate
ML	Machine Learning
CDS	Cryptojacking Detection System
PMCC	Product Moment Correlation Coefficient
FNR	False Negative Rate
TNR	True Negative Rate

CHAPTER 1

INTRODUCTION

1.1 Introduction

Cryptocurrency is quite popular or widespread in our daily lives nowadays. A cryptocurrency is a digital or virtual currency that is safeguarded by cryptography [1], making counterfeiting nearly difficult. Aside from that, cryptocurrency is a good service payment mechanism that may be used for online exchange. Since the blockchain is a decentralised network technology that may distribute on many devices to handle and record these transactions, it can be claimed that cryptocurrencies employ blockchain technology, which is also a part of security technology. The most well-known sort of cryptocurrency, for example, is Bitcoin [2],which is always present in our daily lives.

Following that, a slew of issues such as cryptojacking arose as a result of the growing popularity of bitcoin. The cryptojacking hacker either uses the user's computer for unlawful cryptocurrency mining or is an emerging online danger that hides on the computer and leverages the machine's resources to help them mine. The victims are frequently unaware that the hacker is using their computer to mine cryptocurrency because the crypto mining code runs in the background, making the victim difficult to discover.

Additionally, there are two main ways for hackers to covertly mine cryptocurrency on the victim's device. The hacker's initial strategy is to trick the victim into clicking a dangerous link in an email, at which point they will download the crypto mining code and conduct the mining on the victim's laptop, computer, or any other device [3]. The second technique includes hackers infecting websites or online advertisements with JavaScript code, which is automatically loaded and executed by the victim's browser. To maximise their revenues, hackers always combine the two methods. The script is running on the victim's device while a challenging mathematical problem is being solved, and the solution is then sent to the hacker's server. The strategy used to solve this problem in this project is to use a machine learning

model to detect cryptojacking traits and obtain results in terms of accuracy, true positive rate, and other metrics.

1.2 Background of the Problem

Crypto mining via web browsers and installable binary crypto mining are the two most popular attacks. Asm.js and WebAssembly, two advances in web technology, are used to construct web pages utilizing JavaScript technology for browsers based on crypto mining attacks. In addition, 9,000 Internet cafes across China have had their computers taken over by an organized crime group. After searching 853,936 popular online pages, 868 of which are among the top 100,000 websites according to Alexa's ranking, it finds 2770 distinct cryptojacking samples [4]. Experts in law enforcement and cybersecurity have concurred in recent years that cryptojacking is declining. This is a result of numerous massive mining botnet incursions, including a massive botnet that contained roughly 850,000 computers and was discovered and taken offline by a joint operation of the French police and security firm Avast, or a Smominru attack that cryptojacked 500,000 computers to mine cryptocurrency [5].

Since there is presently no anti-cryptojacking attack software or antivirus, it is crucial to stop the rise in cryptojacking attacks as they represent a novel attack. In order to prevent future cryptojacking efforts on the internet, our research will use machine learning to predict new cryptojacking attempts as well as to detect present cryptojacking attacks.

Besides that, the problem in this project is that the feature is too many so that it will decrease machine learning detection. Therefore, this study used a PMCC heatmap and random forest to select the best feature to help machine learning differentiate cryptojacking attacks from normal traffic.

1.3 Objective

There are three objectives in this project which are:

1) To study feature selection of Product Moment Correlation Coefficient (PMCC) algorithm with Heatmap for machine learning model classification and development.
2) To develop a cryptojacking detection system with Product Moment Correlation Coefficient (PMCC) with Heatmap Intelligent.
3) To evaluate the detection performance of the cryptojacking detection system.

1.4 Scope

The scope of this project which are:

1) To focus on detecting cryptojacking attack dataset by using machine learning classification.
2) To do the feature selection using the PMCC Heatmap.
3) To evaluate the result accuracy after detecting the cryptojacking attack datasets using random forest classification.

1.5 Thesis Organization

Chapter 1 is to discuss about the introduction of the project which are introduction, background, problem statement, objectives, project scope and thesis organization.

Chapter 2 is to discuss the literature review of the project. This chapter are focuses on three related works, and compares the expected existing related technology, method is related for this project.

Chapter 3 is to discuss the methodology of the project. This chapter introduces the proposed design, hardware and software specifications.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In this chapter was explains about the related work system in this study. Additionally, this study has covered the feature selection in detail along with each associated work. Based on related research, it also aids in our ability to comprehend various machine learning classifications, such as the advantages and disadvantages of precision.

2.2 Three Related Work on Comparing Existing System

2.2.1 Feature Selection Algorithm and Machine Learning Model

Ransomware has emerged as one of the most prevalent categories of cybercrime as a result of the sharp increase in malware prevalence in recent years. Because ransomware can spread automatically across the Internet. In 2017, we are seeing more instances of it than ever before. In an attempt, this article uses SDN and machine learning to detect ransomware. [6]

Since the Internet is now accessible everywhere, we employ Web services to accomplish a variety of goals. As a result, users of the Internet are seriously threatened by the emergence of dangerous websites. Malicious websites that host undesirable content, such as spam, phishing, drive-by downloads, etc., may prey on unaware people. Moreover, phishing is a common form of online attack. As a result, the author of this research uses feature selection and machine learning to detect phishing. [7]

The Android market share reached 85.3% by the end of 2016 according to IDC's 2016 report. Android has grown more and more necessary in our daily lives thanks to its open-source features and free benefits. Consequently, the quantity of harmful software is continuously

increasing. The question of how to detect great accuracy is therefore a hot topic. An Android malware detection system based on machine learning is the third related work paper. [8]

Table 2.1 The comparison of algorithm and models between existing system

Content	**Feature Selection Algorithm**	**Machine Learning Model**	**Detection**
Paper 1	Kernel-Based Learning Algorithm	Random Forest	Ransomware
Paper 2	InfoGain	J48 Algorithm	Phishing
Paper 3	PCA	SVM	Android Malware
This paper	PMCC heatmap	Random Forest	Cryptojacking

2.2.2 Comparison Features of Related Existing System

Bitdefender is one of the most technologically advanced antivirus programmes available. This software guards against dangerous malware and spyware in addition to viruses. The cloud-based scanner from Bitdefender detects malware with 100% accuracy by combining traditional signature-based scanning with machine learning. In addition, Bitdefender offers some of the best parental controls, a quick VPN, a thorough system tune-up, and powerful online defenses.

TotalAV Antivirus uses heuristic algorithms and proactive defense to completely remove viruses and safeguard users from various malware types. This antivirus programme will routinely update itself with the latest recent definitions. As a result, the user will always be protected from malware such as rootkits, adware, spyware, Trojan horses, and viruses. In order to stop hackers and viruses from accessing the operating system through these back doors, TotalAV provides a private firewall that blocks all connections not started by an authorized programme or process.

McAfee has been providing outstanding antivirus solutions for over three decades and is one of the most established antivirus products available. It offers a variety of scans in addition to real-time virus protection against various threats. Users of the McAfee antivirus software can automate their protection measures and still be safe.

Table 2.2 The comparison of features between existing system

Content	**System 1 (Bitdefender)**	**System 2 (TOTALAV)**	**System 3 (McAfee)**	**This system**
Platform	Windows, Mac OS, iOS, Android devices and Linux	Windows, Mac OS, iOS and Android devices	Windows, Mac OS, iOS, Android devices and Linux	Web-based
Detection capabilities	Excellent	Excellent	Good	Good
System performance	Excellent	Excellent	Good	Good
Value for money	Excellent	Good	Fair	Fair
Detection	Ransomware, phishing and fraud	Phishing and fraud	Android Malware, spyware and adware	Cryptojacking

2.3 Comparative Analysis

"Machine Learning-Based Detection of Ransomware using SDN" is the first similar research project. With the exception of the list of components that make up the SVM, it is challenging to comprehend what the kernel-based learning algorithm has learned, which has the disadvantage of being less effective because it can train on either a few samples or many features. The random forest is used in machine learning categorization. The fact that it is built on a bagging algorithm and makes use of integrated learning technology is a benefit. On the data subset, it generates as many trees as it can before combining their result [9]. The disadvantage is that random forest requires a lot of time to train because it generates multiple trees and bases judgements on the majority of votes. This can not only alleviate the over-fitting issue in the decision tree but also reduce variance and enhance accuracy. [10]

The next related work, Phishing Detection Based on Machine Learning and Feature Selection Methods, the J48 algorithm is used in machine learning classification and InfoGain is used to choose features. For feature selection, the ID3 algorithm employs the information gain criterion. Noise has no effect on the advantage, and instances without attribute values can still be trained [11]. The downside is that overfitting and missing values are not taken into account. An open-source Java implementation of the straightforward C4.5 technique really J48 [12]. Its ability to manage continuous and discrete properties is a benefit. However, while processing continuous attributes, it first generates a threshold and then separates the list of attribute values into two groups: those that are greater than the threshold and those that are lower than or equal to the threshold. Due to C4.5's construction of empty branches, the tree will grow larger and become more intricate. The dataset has a significant number of values that are zero or very close to zero, which means that the tree decision will grow more complex because these values do not contribute to the creation of rules or classes for classification tasks. [13]

Then, the third related work is about the detect an android malware system based on machine learning. The feature is chosen using the Principal Component Analysis (PCA) algorithm, and the classification is done using support vector machines (SVM). Additionally, PCA feature selection has the advantage of enhancing algorithm performance. The algorithm's performance will be impacted severely if there are numerous features [14]. As a result, PCA can eliminate pertinent variables that are not used in accelerated machine learning algorithms'

decision-making processes. As a result, it will cut down on the algorithm's feature count and training time. In addition, PCA has the disadvantage of resulting in information or data loss. Although the primary components make an attempt to account for as much feature diversity as possible in the dataset. Additionally, information will be lost if the number of principal components is unintentionally chosen in contrast to the original feature list. The benefit of SVM is that it can perform better and be more efficient in high-dimensional space when there is a clear separation margin between classes. SVM's inability to handle huge data sets is another drawback of the technique. When the classes overlap, performance decreases when the data set contains numerous noisy sample targets. [15]

As for Bitdefender, the system includes advantages and disadvantages for each component. Bitdefender does not alter the price of its plans and products, which is a factor in value for money. The consumer had received a guarantee from Bitdefender regarding the brand's features and services. The high price rate, however, was a drawback for Bitdefender. The user had noticed a significant difference in features between the highly rated Bitdefender product and the version with lower ratings. The fact that Bitdefender barely interferes with other programmes gave it an advantage in terms of system performance. This implies that Bitdefender's operation speeds up system performance. However, Bitdefender has certain drawbacks as well, slowing down system performance by taking longer than usual to scan the entire system. Bitdefender might use more power while scanning the entire platform, which might reduce battery life. Bitdefender provides the most complete features for both identifying and eliminating malware in the area of malware detection. Bitdefender has the fastest malware detection rates and continuously monitors the platform's health.

On the other hand, TotalAv has advantages and disadvantages for each aspect. TotalAV has the benefit of requiring less memory and system resources, which means the device's performance won't be slowed down in terms of system performance. TotalAV is always running in the background looking for malicious attacks. It has a disadvantage, nevertheless, in terms of system performance: TotalAV requires more time to run a full system scan, which might take up to an hour depending on the size and usage of the hard drive. For individuals who want to finish the scan fast, it is inconvenient. TotalAV has a significant issue when it comes to malware detection because the free edition does not offer real-time antivirus protection. Only by performing a manual system scan can the user identify malware risks.

In contrast, McAfee has advantages and drawbacks in each category. Based on the subscription cost following the free trial, McAfee's value for money is determined. Depending on the equipment and licencing the user wants, it will be expensive. McAfee's email scanning tool checks all email attachments and HTML content to ensure that no malicious malware was downloaded, which improves system efficiency. McAfee, however, has a performance issue; it is incapable of doing thorough device scans. It took over an hour to finish the thorough scan. The scanning engine of McAfee identifies and eliminates malware when it comes to malware detection. It offers locate-and-fix answers to all dangers, both known and unknown. But McAfee had a limit in detection. The programme from McAfee could be unable to identify Trojans.

With the use of machine learning's random forest and PMCC heatmap feature selection, a cryptojacking detection system is being developed in this study. The PMCC heatmap's feature selection benefit is that its conceptual structure is simple to comprehend, analyse, and resolve. As a result, if there is any relationship between X and Y, each change in X must result in a change in Y that is proportionate to the change in X; if the relationship is not linear, the result will be erroneous. The bagging method, which uses integrated learning technology and has the highest accuracy, is another benefit of random forest. The limitation of random forest classification is that it takes longer to train and obtain results.

Table 2.3 The comparison advantage and disadvantages of related work

Content	Feature Selection Algorithm		Machine Learning Model	
	Advantage	**Disadvantage**	**Advantage**	**Disadvantage**
Paper 1	More effective	Incomprehensibility	Bagging algorithm, utilizes integrated and highest accuracy.	Spend time to train.
Paper 2	Not affect by noise and instances lack attribute value also can be trained.	Does not consider missing value and overfitting issues.	Can handle continuous and discrete attributes.	Making the tree bigger and more complex
Paper 3	Improve the algorithm performance.	Information will loss.	More effective in high-dimensional space.	Not suitable for large data sets.
This paper	Concept easy understands, interpret and solve.	It only measures the linear relationship between X and Y, and if there is any relationship, any change in X must have a constant proportional change in Y.	Bagging algorithm, utilizes integrated and highest accuracy.	Spend time to train.

2.4 Chapter Summary

The comparison of the features and specifications of three comparable works in this chapter marks the conclusion of the literature review. The three pertinent studies selected are machine learning-based ransomware detection utilizing SDN, phishing detection using machine learning and feature selection techniques, and identifying an Android malware system using machine learning. Three aspects of the feature selection technique and machine learning model comparison are discussed. Three relevant existing systems—Bitdefender, TotalAV, and McAfee—were chosen for the system selection. The attributes of the systems are contrasted and expressed in a comparative analysis. The advantages and disadvantages of each choice are also covered in this chapter.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter explains the system's overall methodology and framework. To ensure the project's smooth development, this technique model is employed during project development. Development methodology describes a collection of methods or procedures used to create machine-learning random forest classification and feature selection. In addition, the system completes the development term utilizing the Python programming language. The proposed project architecture, the random forest classifier algorithm, the web application for the cryptojacking detection system, and the feature selection technique for PMCC heatmaps will all be covered in this study.

3.2 Proposed Design & Interface

The below shows the flow chart of the project. The development of the PMCC Heatmap and Random Forest machine learning classifier for detecting cryptojacking has three stages. Additionally, the cryptojacking attack time series dataset will be used to create the dataset. The literature review phase is the first, the model development phase is the second, the outcome analysis phase is the third, the web application interface development is the final phase.

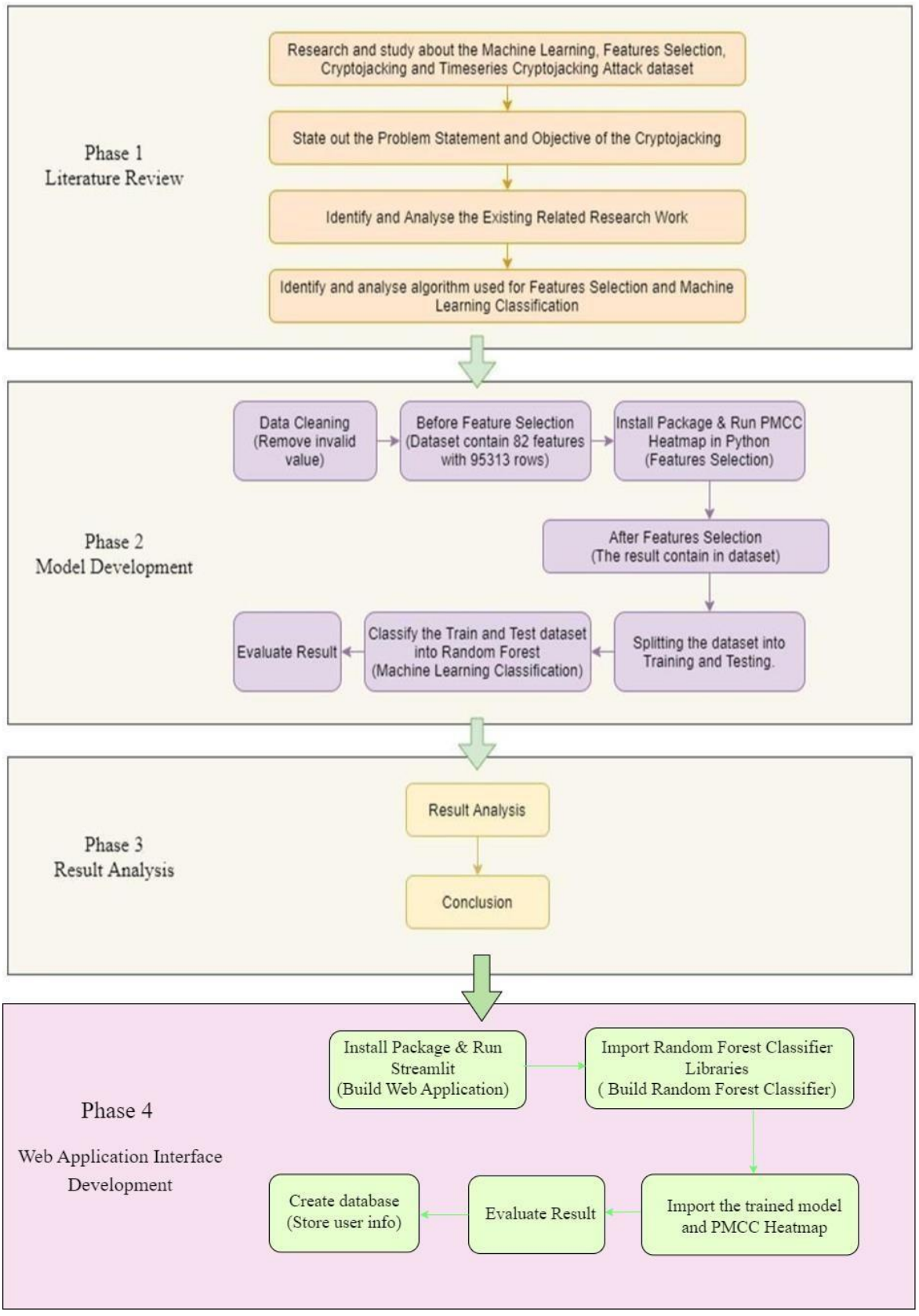


Figure 3.2.1 The flow chart of the project

3.2.1 PMCC Heatmap

The Product Moment Correlation Coefficient (PMCC) Heatmap is utilized to process the data and creates the heatmap. However, data cleaning must be completed and any invalid and noise data in the dataset will be eliminated before the PMCC heatmap can be used. Determine the dataset feature that will become the variable Y. The output of the correlation coefficient based on each feature will then be printed by the system. We can easily grasp and picture the outcome. Setting the threshold to obtain the best feature based on the algorithm will be the following stage. Additionally, if the users require additional testing, they can establish a new criterion to obtain the greatest functionality. The system will finally print out the best features based on the threshold that sets in previously.

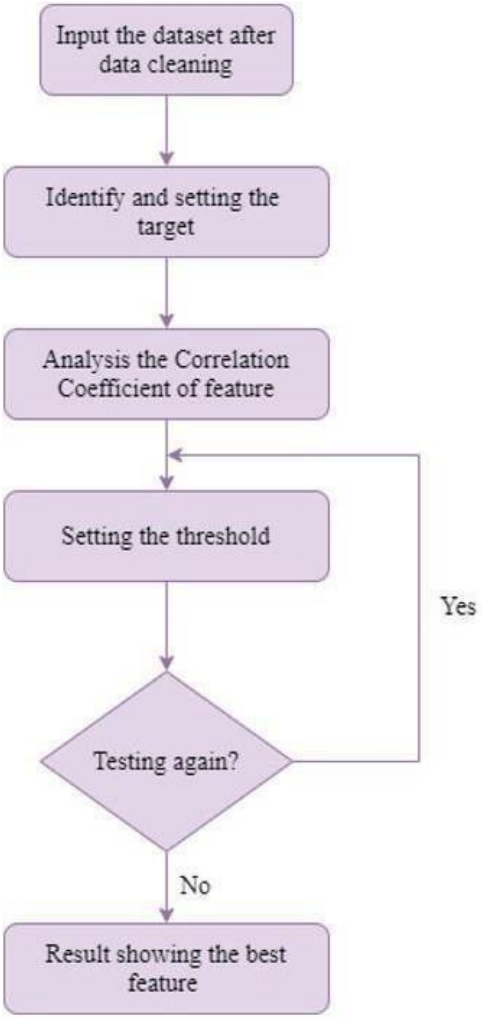


Figure 3.2.2 The flow chart of PMCC heatmap

3.2.2 Random Forest Algorithm Classification

The best feature from the dataset will be chosen to begin the classification process in the random forest method. The algorithm will then build a decision tree for each feature and use each tree's prediction result. Every anticipated outcome will be executed by the system, which will vote. In final stage, the system will select the prediction outcome that receives the most votes as the final prediction outcome.

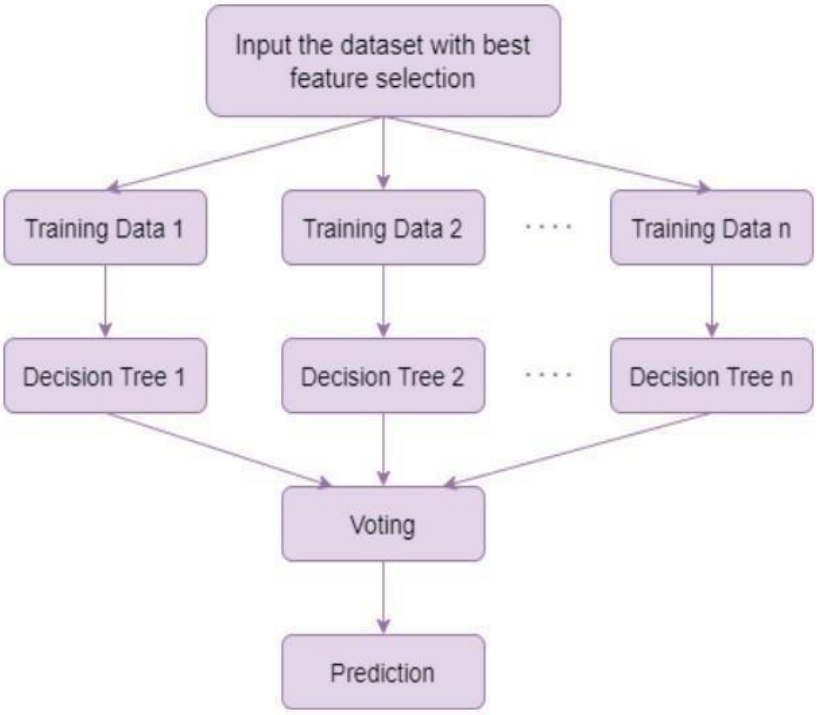


Figure 3.2.3 The flow chart of random forest classification

3.2.3 Cryptojacking Detection System Web Application Interface

The web application interface for the Cryptojacking Detection System is shown below. The user can input the data into each respective feature. The data inserted will be displayed in the table. The result of the prediction and heatmap will be auto-generated and displayed to the users after they inserted the data. The cryptojacking detection system web application was created using streamlit python and detailed explanation on the structure of the web application is discussed further in Appendix B and Appendix C.

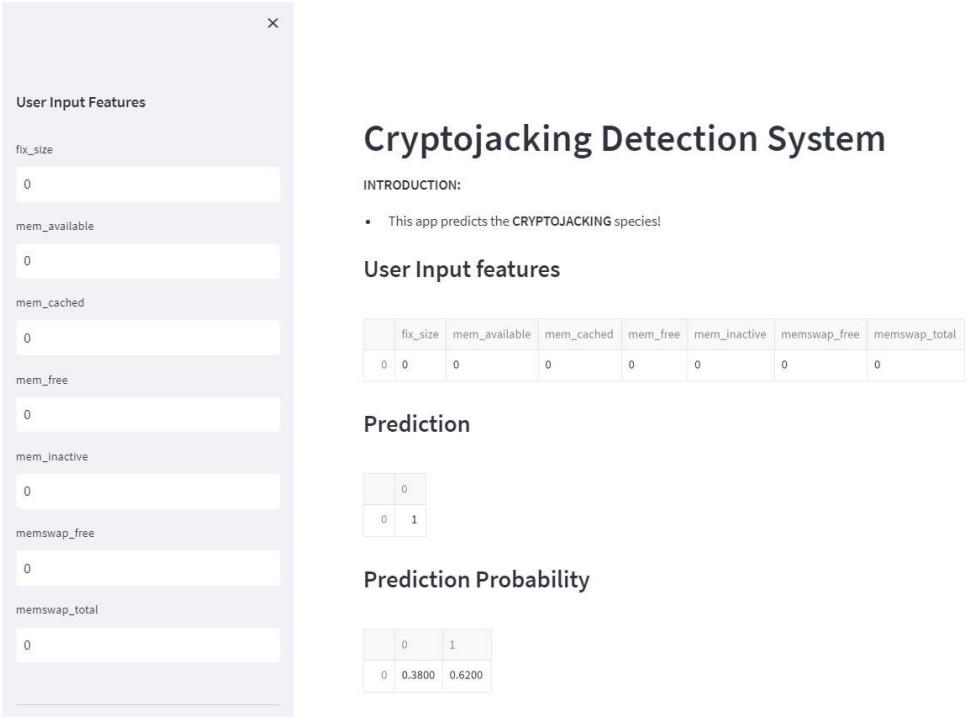


Figure 3.2.4 The interface of the cryptojacking detection system web application

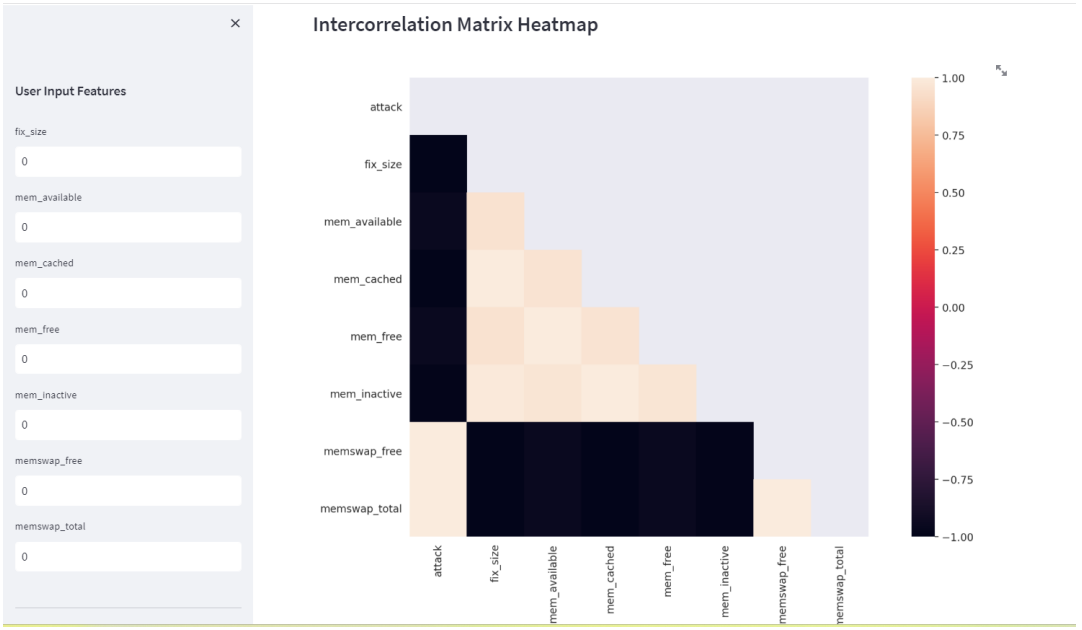


Figure 3.2.5 The heatmap of the cryptojacking detection system web application

3.3 Methodology

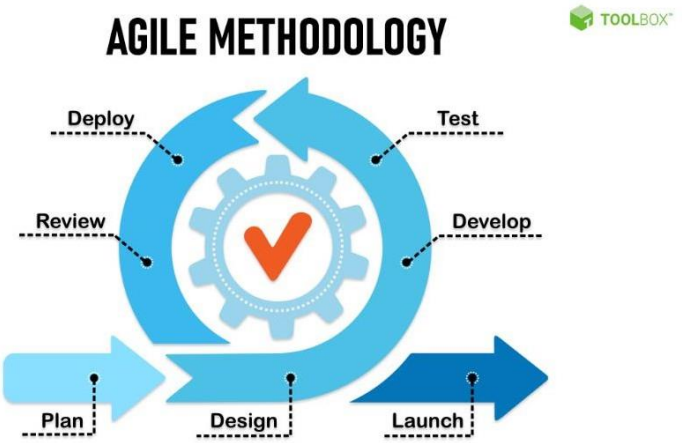


Figure 3.3.1 Agile model

The Software Development Life Cycle (SDLC) employed in this project is an agile methodology. Iterative and incremental processes are both included in the definition of the agile model. Requirements analysis, design, development, testing, and deployment are the five stages that make up this process. This project adopts agile methodology as the development process for the Cryptojacking Detection System. Agile methodology is chosen because it exemplifies quick, adaptable responses to change. Agile development helps in changing functions and needs for the system's improvement because there is a limited amount of time to build the project and the requirements are not fully and accurately solicited at first. Additionally, as the project was being developed, some needs were added and changed, and agile allows the project the flexibility to incorporate the new modifications into it without much risk.

The first phase, the requirement phase, collects the requirements for the proposed project. The system's functions and requirement scope are extracted and identified during the requirement phase. The primary goal of this project is to create an application that can detect the risk of cryptojacking on a user's device. In this stage, the datasets used for cryptojacking are gathered, and data pre-processing is carried out. This Cryptojacking Detection System was developed after careful analysis and comparison of existing anti-virus software and systems. In this stage, the project's modules and functionalities are gathered and considered. Moreover, Software Requirement Specification (SRS), which

includes a context diagram, use case diagram, and its description, is proposed to define how Cryptojacking Detection System is constructed.

The visual design is then proposed throughout the design process by creating the system's user interface. This helps in determining the appropriate hardware and software specifications and also how the system architecture should be built. Additionally, the Cryptojacking Detection System prototype interfaces have been created. The system architecture, detailed design, and data dictionary of the system are also reviewed from the Software Design Documentation (SDD).

The Python Streamlit platform is used to develop the Cryptojacking Detection System during the development phase. Other libraries are imported to make sure all the requirements are met and the system functions properly. Streamlit Python provides a reliable web framework for developing this application.

In the testing phase, each system function is tested during this stage, which is followed by testing the overall system flow. Before moving on to the following phase, any errors or bugs will be resolved.

The last stage of this Agile approach is deployment. The Cryptojacking Detection System is deployed via an AWS E2 server to the actual environment and is fully operational without any issues.

In addition, the Agile cycle would be repeated until satisfaction is reached if the system is not ready to be deployed.

3.4 Dataset Collection and Analysis

The dataset used in this study for cryptojacking was gathered from the top 1 million Alexa-ranked websites. Before being used for testing, the dataset had to be cleaned up since it was in raw data form.

65% of the dataset is allocated for training and 35% for testing. The first 100 projects' datasets were used to train the system to detect cryptojacking, while the remaining 30 projects' datasets were used to evaluate the system's performance.

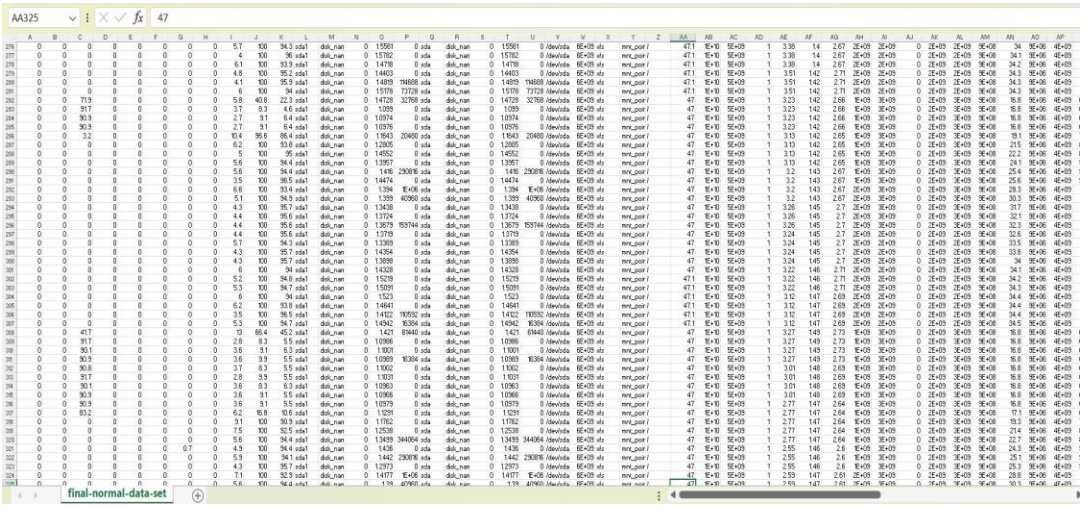


Figure 3.4.1 Uncleaned dataset (final-normal-data-set)

The normal datasets include the time-series performance data during no cryptojacking attacks.

Source: (Kaggle's: Cryptojacking Attack Timeseries Dataset.)

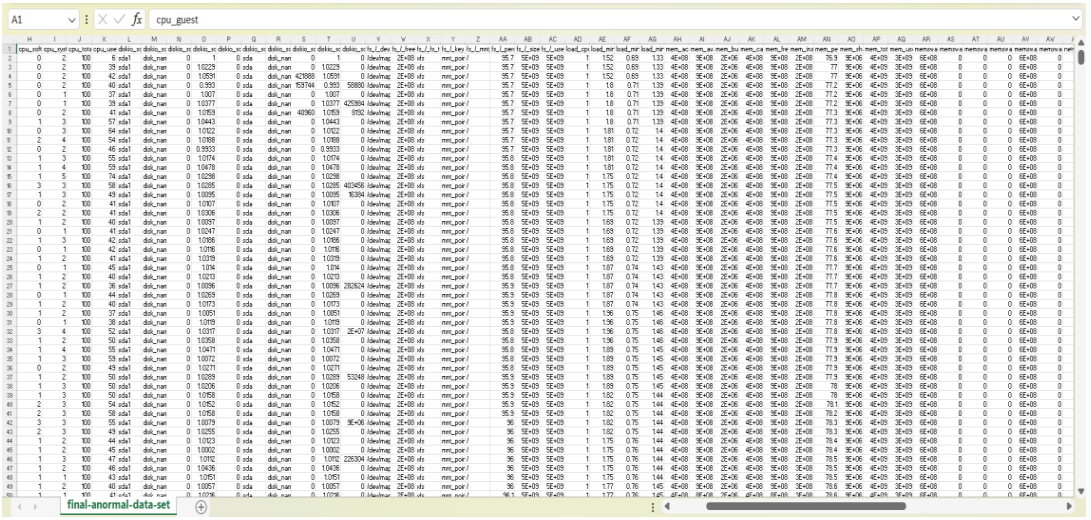


Figure 3.4.2 Uncleaned dataset (final-anormal-data-set)

The anormal dataset includes the time-series performance data during a cryptojacking at attack.

Source: (Kaggle's: Cryptojacking Attack Timeseries Dataset.)

3.5 Hardware and Software Specifications

Table 3.1 The hardware and software specifications

Name	Version	Type	Description	Purpose
Laptop	LENOVO IdeaPad 3 15ALC6	Hardware	Portable personal computer	Used to develop and run the software.
Python	3.9	Software	Python is an explanatory, object oriented, high-level programming language with dynamic semantics.	Used to edit and develop the python languageto create the algorithm.
Jupyter Notebook	6.3.0	Software	Jupyter notebook is a web-based interactive development environment for notebooks, code, and data.	Used to test and develop the system. This software can helpto get the result while running the system.
Microsoft Office	2021	Software	Microsoft Office is a set of applications designed to help improve work efficiency and complete common tasks on your computer.	Used for drawing ,complete thereport in this project.
Google Chrome	100.0.4896.88	Software	Google Chrome browser is an open source program for accessing the World Wide Web and running Web-based applications.	Used to search related information and references for the project.
GitHub	2.9.0	Software	GitHub is a code hosting platform for version control and collaboration. It allow user to work together from anywhere.	Used to upload and store codingof the project.

3.6 Chapter Summary

The development process, the PMCC heatmap feature selection technique, the random forest classifier algorithm, and the web application for the cryptojacking detection system have all been covered in detail in this chapter. The development was carried out using Streamlit Python. During the design phase, the project's entire proposed design is explained. The sources of the crytojacking datasets are also mentioned above. Last but not least, the hardware & software specification includes a list and an explanation of all the information pertaining to the hardware and software used for the development of this study.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

This chapter will discuss the process of development of the cryptojacking detection application using PMCC heatmap and the testing result. Streamlit Python is mainly used to develop the web frame and interface of cryptojacking detection system in this project. In addition, this chapter will show all the interfaces and results of the cryptojacking detection system.

4.2 Result

The results of the PMCC heatmap and the random forest machine learning classification are shown in this sub-topic.

4.2.1 PMCC heatmap

The PMCC heatmap of cryptojacking attack timeseries dataset is shown in the figure below. The correlation heat map, which shows the correlation between various variables, is a graphical depiction of the correlation matrix.

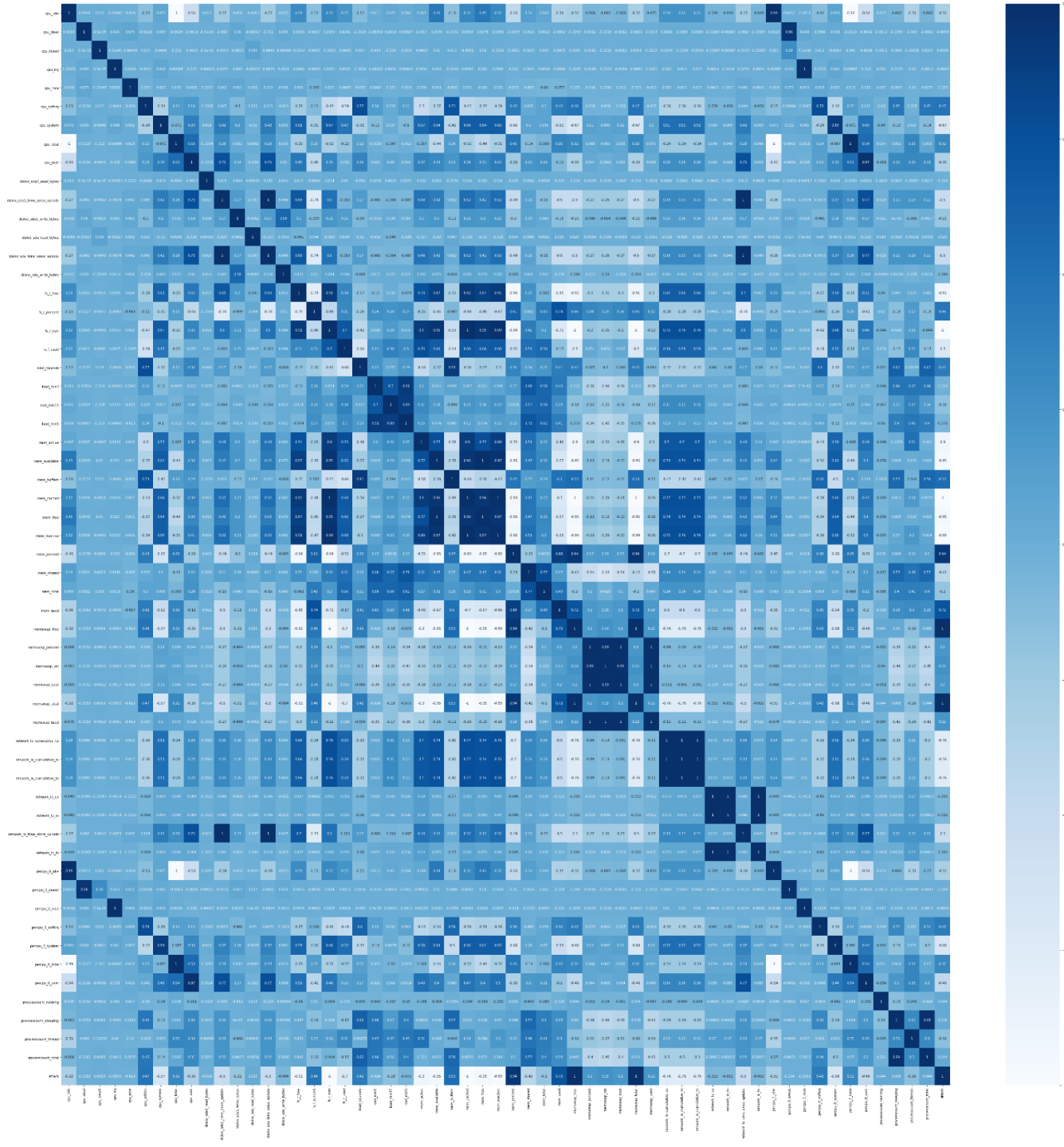


Figure 4.2.1.1 PMCC heatmap of crytojacking attack timeseries

The table below shows the result of the number of best features based on thethreshold correlation on the PMCC heatmap.

Table 4.2.1 The result of the number of the features based on the threshold correlation.

Threshold Correlation	Number of Attribute
<0.05	12
<0.1	15
<0.2	18
<0.3	23
>0.4	24
>0.5	17
>0.65	15
>0.75	11
>0.85	10
>0.95	7

4.2.2 Random Forest Classification

The results of the random forest classification based on the selection of heatmap correlation features with threshold correlations of 0.05, 0.1, 0.2,
0.3, >0.4, >0.5, >0.65, >0.75, >0.85 and >0.95 are shown in the figure below. In this experiment, I divided the data so that 65% would be used for training in the random forest classification and 35% would be used for testing. The random forest classification experiment is then run with the parameter's default value.

```
Train Result

Accuracy Score: 94.79%

Classification Report:
                    -1            1  accuracy     macro avg  weighted avg
precision     0.958174     0.938127  0.947867      0.948150      0.948097
recall        0.936036     0.959572  0.947867      0.947804      0.947867
f1-score      0.946975     0.948729  0.947867      0.947852      0.947857
support    9349.000000  9449.000000  0.947867  18798.000000  18798.000000

Confusion Matrix:
 [[8751  598]
 [ 382 9067]]

--------------------------------------------------------------------------------

Test Result

Accuracy Score: 93.20%

Classification Report:
                    -1            1  accuracy     macro avg  weighted avg
precision     0.943809     0.920607  0.932029      0.932208      0.932323
recall        0.920172     0.944123  0.932029      0.932148      0.932029
f1-score      0.931841     0.932217  0.932029      0.932029      0.932027
support    5111.000000  5011.000000  0.932029  10122.000000  10122.000000

Confusion Matrix:
 [[4703  408]
 [ 280 4731]]
```

Figure 4.2.2.1 show the result of classification with feature are <0.05 in threshold correlation

```
Train Result

Accuracy Score: 99.96%

Classification Report:
                    -1            1  accuracy     macro avg  weighted avg
precision     0.999254     1.000000  0.999628      0.999627      0.999628
recall        1.000000     0.999257  0.999628      0.999628      0.999628
f1-score      0.999627     0.999628  0.999628      0.999628      0.999628
support    9377.000000  9421.000000  0.999628  18798.000000  18798.000000

Confusion Matrix:
 [[9377    0]
 [   7 9414]]

--------------------------------------------------------------------------------

Test Result

Accuracy Score: 99.33%

Classification Report:
                    -1            1  accuracy     macro avg  weighted avg
precision     0.992149     0.994430  0.993282      0.993290      0.993285
recall        0.994491     0.992062  0.993282      0.993277      0.993282
f1-score      0.993319     0.993245  0.993282      0.993282      0.993282
support    5083.000000  5039.000000  0.993282  10122.000000  10122.000000

Confusion Matrix:
 [[5055   28]
 [  40 4999]]
```

Figure 4.2.2.2 show the result of classification with feature are <0.1 in threshold correlation

```
Train Result

Accuracy Score: 100.00%

Classification Report:
              -1       1  accuracy  macro avg  weighted avg
precision     1.0     1.0       1.0        1.0           1.0
recall        1.0     1.0       1.0        1.0           1.0
f1-score      1.0     1.0       1.0        1.0           1.0
support    9405.0  9393.0       1.0    18798.0       18798.0

Confusion Matrix:
 [[9405    0]
 [   0 9393]]

----------------------------------------------------------------

Test Result

Accuracy Score: 100.00%

Classification Report:
              -1       1  accuracy  macro avg  weighted avg
precision     1.0     1.0       1.0        1.0           1.0
recall        1.0     1.0       1.0        1.0           1.0
f1-score      1.0     1.0       1.0        1.0           1.0
support    5055.0  5067.0       1.0    10122.0       10122.0

Confusion Matrix:
 [[5055    0]
 [   0 5067]]
```

Figure 4.2.2.3 show the result of classification with feature are <0.2 in threshold correlation

```
Train Result

Accuracy Score: 100.00%

Classification Report:
              -1       1  accuracy  macro avg  weighted avg
precision     1.0     1.0       1.0        1.0           1.0
recall        1.0     1.0       1.0        1.0           1.0
f1-score      1.0     1.0       1.0        1.0           1.0
support    9444.0  9354.0       1.0    18798.0       18798.0

Confusion Matrix:
 [[9444    0]
 [   0 9354]]

----------------------------------------------------------------

Test Result

Accuracy Score: 100.00%

Classification Report:
              -1       1  accuracy  macro avg  weighted avg
precision     1.0     1.0       1.0        1.0           1.0
recall        1.0     1.0       1.0        1.0           1.0
f1-score      1.0     1.0       1.0        1.0           1.0
support    5016.0  5106.0       1.0    10122.0       10122.0

Confusion Matrix:
 [[5016    0]
 [   0 5106]]
```

Figure 4.2.2.4 show the result of classification with feature are <0.3 in threshold correlation

```
Train Result

Accuracy Score: 100.00%

Classification Report:
              -1       1  accuracy  macro avg  weighted avg
precision     1.0     1.0       1.0        1.0           1.0
recall        1.0     1.0       1.0        1.0           1.0
f1-score      1.0     1.0       1.0        1.0           1.0
support    9372.0  9426.0       1.0    18798.0       18798.0

Confusion Matrix:
 [[9372    0]
 [   0 9426]]

------------------------------------------------------------

Test Result

Accuracy Score: 100.00%

Classification Report:
              -1       1  accuracy  macro avg  weighted avg
precision     1.0     1.0       1.0        1.0           1.0
recall        1.0     1.0       1.0        1.0           1.0
f1-score      1.0     1.0       1.0        1.0           1.0
support    5088.0  5034.0       1.0    10122.0       10122.0

Confusion Matrix:
 [[5088    0]
 [   0 5034]]
```

Figure 4.2.2.5 show the result of classification with feature are >0.4 in threshold correlation

```
Train Result

Accuracy Score: 100.00%

Classification Report:
              -1       1  accuracy  macro avg  weighted avg
precision     1.0     1.0       1.0        1.0           1.0
recall        1.0     1.0       1.0        1.0           1.0
f1-score      1.0     1.0       1.0        1.0           1.0
support    9359.0  9439.0       1.0    18798.0       18798.0

Confusion Matrix:
 [[9359    0]
 [   0 9439]]

------------------------------------------------------------

Test Result

Accuracy Score: 100.00%

Classification Report:
              -1       1  accuracy  macro avg  weighted avg
precision     1.0     1.0       1.0        1.0           1.0
recall        1.0     1.0       1.0        1.0           1.0
f1-score      1.0     1.0       1.0        1.0           1.0
support    5101.0  5021.0       1.0    10122.0       10122.0

Confusion Matrix:
 [[5101    0]
 [   0 5021]]
```

Figure 4.2.2.6 show the result of classification with feature are >0.5 in threshold correlation

```
Train Result

Accuracy Score: 100.00%

Classification Report:
              -1       1  accuracy  macro avg  weighted avg
precision     1.0     1.0       1.0        1.0           1.0
recall        1.0     1.0       1.0        1.0           1.0
f1-score      1.0     1.0       1.0        1.0           1.0
support    9359.0  9439.0       1.0    18798.0       18798.0

Confusion Matrix:
 [[9359    0]
 [   0 9439]]

------------------------------------------------------------

Test Result

Accuracy Score: 100.00%

Classification Report:
              -1       1  accuracy  macro avg  weighted avg
precision     1.0     1.0       1.0        1.0           1.0
recall        1.0     1.0       1.0        1.0           1.0
f1-score      1.0     1.0       1.0        1.0           1.0
support    5101.0  5021.0       1.0    10122.0       10122.0

Confusion Matrix:
 [[5101    0]
 [   0 5021]]
```

Figure 4.2.2.7 show the result of classification with feature are >0.65 in threshold correlation

```
Train Result

Accuracy Score: 100.00%

Classification Report:
              -1       1  accuracy  macro avg  weighted avg
precision     1.0     1.0       1.0        1.0           1.0
recall        1.0     1.0       1.0        1.0           1.0
f1-score      1.0     1.0       1.0        1.0           1.0
support    9442.0  9356.0       1.0    18798.0       18798.0

Confusion Matrix:
 [[9442    0]
 [   0 9356]]

------------------------------------------------------------

Test Result

Accuracy Score: 100.00%

Classification Report:
              -1       1  accuracy  macro avg  weighted avg
precision     1.0     1.0       1.0        1.0           1.0
recall        1.0     1.0       1.0        1.0           1.0
f1-score      1.0     1.0       1.0        1.0           1.0
support    5018.0  5104.0       1.0    10122.0       10122.0

Confusion Matrix:
 [[5018    0]
 [   0 5104]]
```

Figure 4.2.2.8 show the result of classification with feature are >0.75 in threshold correlation

```
Train Result

Accuracy Score: 100.00%

Classification Report:
              -1       1  accuracy  macro avg  weighted avg
precision     1.0     1.0       1.0        1.0           1.0
recall        1.0     1.0       1.0        1.0           1.0
f1-score      1.0     1.0       1.0        1.0           1.0
support    9359.0  9439.0       1.0    18798.0       18798.0

Confusion Matrix:
 [[9359    0]
 [   0 9439]]

------------------------------------------------------------

Test Result

Accuracy Score: 100.00%

Classification Report:
              -1       1  accuracy  macro avg  weighted avg
precision     1.0     1.0       1.0        1.0           1.0
recall        1.0     1.0       1.0        1.0           1.0
f1-score      1.0     1.0       1.0        1.0           1.0
support    5101.0  5021.0       1.0    10122.0       10122.0

Confusion Matrix:
 [[5101    0]
 [   0 5021]]
```

Figure 4.2.2.9 show the result of classification with feature are >0.85 in threshold correlation

```
Train Result

Accuracy Score: 100.00%

Classification Report:
              -1       1  accuracy  macro avg  weighted avg
precision     1.0     1.0       1.0        1.0           1.0
recall        1.0     1.0       1.0        1.0           1.0
f1-score      1.0     1.0       1.0        1.0           1.0
support    9431.0  9367.0       1.0    18798.0       18798.0

Confusion Matrix:
 [[9431    0]
 [   0 9367]]

------------------------------------------------------------

Test Result

Accuracy Score: 100.00%

Classification Report:
              -1       1  accuracy  macro avg  weighted avg
precision     1.0     1.0       1.0        1.0           1.0
recall        1.0     1.0       1.0        1.0           1.0
f1-score      1.0     1.0       1.0        1.0           1.0
support    5029.0  5093.0       1.0    10122.0       10122.0

Confusion Matrix:
 [[5029    0]
 [   0 5093]]
```

Figure 4.2.2.10 show the result of classification with feature are >0.95 in threshold correlation

4.3 Analysis of Accuracy Result Random Forest in Training and Testing

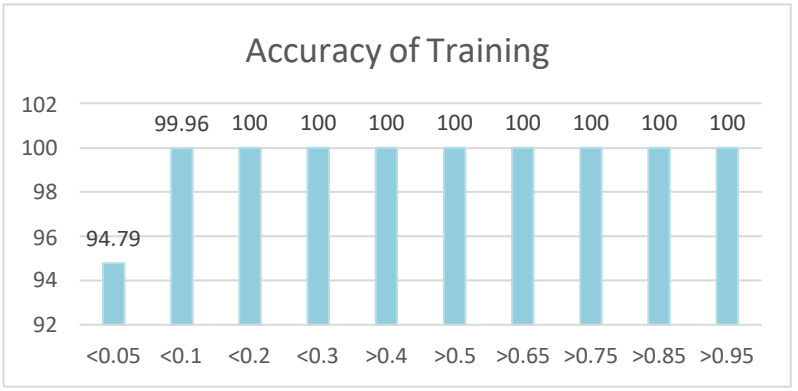


Figure 4.3.1 showing the accuracy of training result in the random forest classification

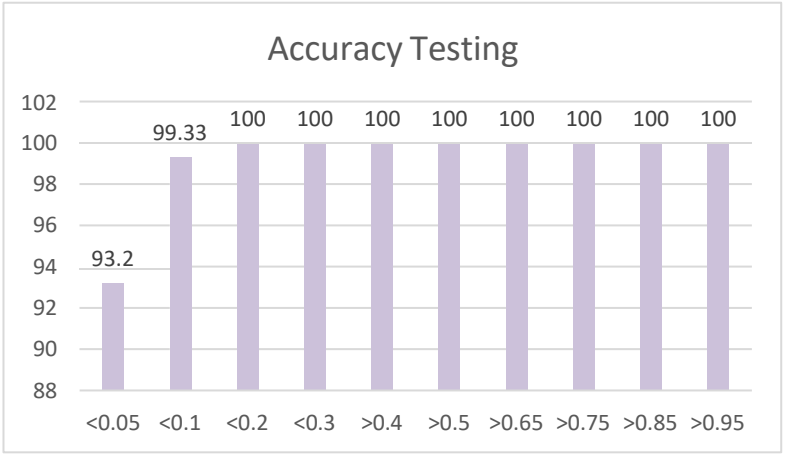


Figure 4.3.2 showing the accuracy of testing result in the random forest classification

Based on the two aforementioned statistics, it is clear that if all of the features have low correlation coefficient values, the accuracy of the results will be affected. For instance, in figure 4.3.1, the threshold 0.05 shows that the training accuracy is just 94.79%, and in figure 4.3.2, the testing accuracy is also at its lowest, at 93.2%. Furthermore, the training accuracy at the threshold of 0.1 is second lowest at 99.96%, while the testing accuracy is second lowest at 99.33%. The accuracy in the threshold ranges of 0.2, 0.3, >0.4, >0.5, >0.65, >0.75, >0.85, >0.95 is 100% as a consequence of training and testing. To conclude, feature selection is crucial because it can increase the effectiveness of our machine-learning efforts to identify cryptojacking.

4.4 Random Forest Classification

Table 4.4.1 Description of parameter and attributes of Random Forest Classification

Parameter & Attributes	Description
n_estimators	The number of trees in the forest.
bootstrap	If the boostrap is True as a default, the bootstrap samples will be using when building trees. Besides that, the whole dataset is used to build each tress when the bootstrap is false.
class_weight	The default value of class_weight is None. The weight associated with a class of the form {class_label: weight}. If not given, all classes should have a weight. For multiple output problems, a list of dictionaries can be provided in the same order as the columns of y.
criterion	To measure the quality of a split. The value of criterion is 'Gini' for impurity and the 'Entropy' is for information gain.
max_depth	The maximum depth of the tree. The default value of max_depth is None and value must be in int value. If the value is none so that the nodes will expand until all the leaves are less than min_sample_split samples.
max_leaf_nodes	The value must be in int and the default value is None. The best node is defined as the relative reduction of impurities. If None then unlimited number of leaf nodes.
min_impurity_decrease	The default value is 0.0 and it must in float number. If this split causes the impurity reduction to be greater than or

	equal to this value, the node will be split.
min_samples_leaf	The value of min_samples_leaf can be int or float but as a default value is using the int value which are 1. The minimum number of samples needs to be located at the leaf node. Any depth of split point will only be considered when there is at least min_samples_leaf training samples in the left and right branches. As a default value, it considers the min_samples_leaf as the minimum number.
min_samples_split	The value of min_samples_split can be in int or float. As a default, the value is 2 in int. If the value is int, then consider the min_samples_split as the minimum number.
min_weight_fraction_leaf	The minimum weighted score of the sum of all input samples required by the leaf node. The default value is 0.0 and must with float input value.
n_jobs	The number of jobs is running in the parallel and the fit, predict, decision_path and apply are all parallelized on the tree.
verbose	To controls the verbosity when fitting and predicting and the default value is 0 and must be in int value.
warm_start	If default value of warm_start is False so it just fit a whole new forest but if set True, it will reuse the solution of the previously to adapt and add more estimators to integration.
oob_score	Use out-of-bag is estimation the general scores. This attribute only exists when oob_score is True.

4.4.1 Confusion Matrix

The confusion matrix is a performance measure for machine learning classification experiments or problems, where the output can be two or more classes. This is a table containing 4 different combinations of predicted and actual values. The 4 different combinations are True Positive (TP), False Positive (FP), False Negative (FN), True Negative (TN). The interpretation of True Positive is the classification are predicted positive and it's true. The example is classification predicted that attack is crytojacking attack and it actually is. Besides that, the interpretation of True Negative is the classification are predicted negative and it's true. The example is classification predicted that attack is not crytojacking attack and it actually is not. Next, the interpretation of False Positive is the classification are predicted positive and it's false. The example is classification predicted that attack is crytojacking attack and it actually is not. Lastly, the interpretation of False Negative is the classification are predicted negative and it's false. The example is classification predicted that attack is not crytojacking attack and it actually is.

Table 4.4.1.1 The confusion matrix of TP, FP, FN, TN

	p' (Predicted)	**n' (Predicted)**
p (Actual)	**True Positive**	**False Negative**
n (Actual)	**False Positive**	**True Negative**

4.5 Implementation

In this section, the first part of this system is the development environment and the second part is the system functionality. All the system functionality will be explained in detail below.

4.5.1 Development environment

The cryptojacking detection application is a dynamic web-based system. It is developed by using Python Streamlit which is an open-source application framework. Python streamlit provides the framework that makes it easier to design the user interface. Moreover, Anaconda Powershell Prompt is used to conduct and set up the environment for running python's library.

4.5.2 System functionality

This section describes system functionality and how the system will interact with users. First, the interface of the system, an interface should be simple and understandable for the user to enable the user to have quick access to the system. The overall application will be explained in more detail below.

a) Dashboard

Figure 4.5.2.1 show the dashboard of Cryptojacking Detection System. The sidebar at the left shows the user input features which enable users to input data into the system. The prediction column is the result produced from the system after the user had input the data into the system. The prediction probability column shows the probability of the prediction.

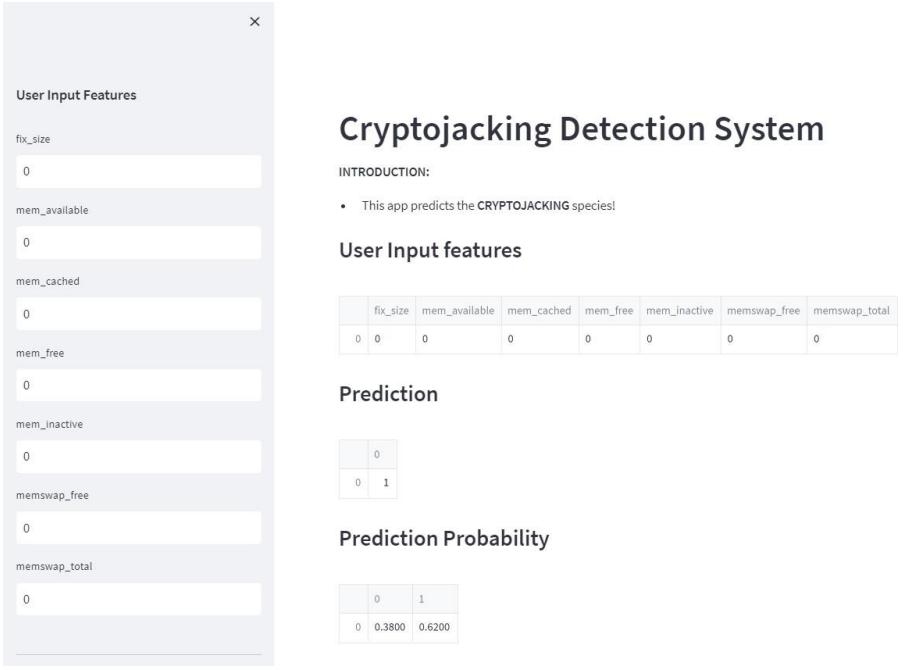


Figure 4.5.2.1 Dashboard of Cryptojacking Detection System

4.6 Testing and result discussion

A testing process is necessary after developing the application to check the usability and quality of the system. Hence, two strategies which are unit testing and user acceptance testing (UAT) are utilized in testing this application. Unit testing is used to check the functionality of each unit while UAT is carried out to test the functionality and ensure application requirements meet user needs. Testing will keep running during the development stage.

4.6.1 Unit testing

Unit testing is being conducted for each component in the Cryptojacking Detection System. Every component is tested after it has been completed. Hence, the objective of unit testing is to test the aspects of the internal components without interaction with another component. This is to ensure that the unit testing objects fulfilled the specified functionality thoroughly and accurately.

4.6.2 User Acceptance Testing (UAT)

This UAT is being conducted after the Cryptojacking Detection System is deployed. The objective of UAT is to let end users access its functionality for the intended use. The system's capability for deployment in a real environment will be determined by the end user. Additionally, a survey of other users has been carried out to determine whether the system satisfies their needs. The UAT document can be referred to in Appendix D.

Discussion on user acceptance testing (UAT)

Based on the UAT that is conducted, the result that can be obtained from the test is as follow. In the user input feature, the user can insert the data successfully and the data inserted are displayed accordingly in the table. The prediction column and probability prediction column also show the accurate result to the user upon the user inserting data into the system.

In short, the system functionality is good and fulfills the user’s needs. Based on the UAT, users had given some suggestions on improving the system, such as making the system a real-time system, adding more and interesting features, making the system into a mobile application that will be more convenient and useful for society.

4.7 Chapter Summary

Through using cryptojacking attack timeseries dataset, the results of the PMCC heatmap and random forest classification are generally tested in this chapter to identify cryptojacking. The system's development has also been examined, and the system's functionality overall is great. The UAT also collects the user's comment on the system.

CHAPTER 5

CONCLUSION

5.1 Introduction

This chapter includes section 5.2 where conclusions are made for the cyrptojacking detection application. Section 5.3 describes the constraints and limitations of developing this application. Section 5.4 discusses the future work that can be added to make this application more advanced and commercialized. Section 5.5 is about the summary of this chapter.

5.2 Conclusion

Cryptojacking detection application is an application developed to detect potential threats in our devices. In the development phases, data pre-processing is needed to generate a complete dataset to train the model. As for methodology, Random Forest Classifier is selected to be used to train the dataset using PMCC Heatmap and apply to the most accurate model to detect the cryptojacking. As for the last part of development, Phyton Streamlit is used to develop the web frame and interface of the application. GoDaddy is the web-hosting service used to publish this application.

The main idea of developing this application is to increase cyber-attack awareness among public users. As we all know, cyber threats are all around the world, anyone could be the victim of a cyber-attack. Due to this situation, cryptojacking detection application is introduced to detect the potential threats in the user devices and protect the computing resources from misuse.

Nowadays, there is no related cryptojacking application published in the market. This is due to several reasons such as people don’t recognize what is cryptojacking and some may don’t know the consequences of cryptojacking. So this new application can have a unique marketing value as it is further developed.

5.3 Project constraint

The limitation of this project is shown below:

a) String values are not accepted in the PMCC heatmap

The features selection of heatmap only can accept the feature in integer or float value only. Once the dataset contains the data type in a string value, the system will appear the error message and shows that the string value is not acceptable in the PMCC heatmap.

b) Time constraint

The amount of time for developing this system is due to academic commitment. Time management is very crucial in system development, and time needs to be well managed as it can aid in the delivery of enhanced system output. For this project, the time is equally divided for each phase of development which are data pre-processing, feature selection, training the model using PMCC Heatmap, developing of the web application, and deployment of this application.

5.4 Future work

There are several improvements that can be made in the future for Cryptojacking Detection Applications.

a) Implements alert function which enables the users aware of cyber threats in their devices more efficiently.
b) Add front-end and back-end for the application to make it more attractive and commercialize to the public user.
c) Make the application into a mobile application as people are more likely to use smartphones now. This will gain the value of the application and its usefulness.

5.5 Summary

The following are the inspirational idea for developing this application:

a) There is no crytojacking application in the market nowadays. It has a unique marketing value.
b) To increase cyber-attack awareness among public users.
c) To protect the computing resources of public users.
d) Future development into a mobile application as it is widely used for publics.

REFERENCES

[1] James Royal, Ph.D., Kevin Voigt (March 23, 2021). What Is Cryptocurrency? Here's What YouShould Know. Retrieved from https://www.nerdwallet.com/article/investing/cryptocurrency-7-things-to-know

[2] JAKE FRANKENFIELD (MAR 7, 2021). Cryptocurrency. Retrieved from https://www.investopedia.com/terms/c/cryptocurrency.asp

[3] Michael Nadeau (MAR 11, 2021). What is cryptojacking? How to prevent, detect, and recover fromit. Retrieved from https://www.csoonline.com/article/3253572/what-is-cryptojacking-how-to-prevent-detect-and-recover-from-it.html

[4] Varlioglu, S., Gonen, B., Ozer, M., & Bastug, M. (2020, March). Is cryptojacking dead after coinhive shutdown?. In 2020 3rd International Conference on Information and Computer Technologies (ICICT) (pp. 385-389). IEEE.

[5] Caprolu, M., Raponi, S., Oligeri, G., & Di Pietro, R. (2019). Cryptomining makes noise: a machine learning approach for cryptojacking detection. arXiv preprint arXiv:1910.09272.

[6] Cusack, G., Michel, O., & Keller, E. (2018, March). Machine learning-based detection of ransomware using SDN. In Proceedings of the 2018 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization (pp. 1-6).

[7] Conference Proceedings (Vol. 1864, No. 1, p. 020136). AIP Publishing LLC.Almseidin, M., Zuraiq,
A. A., Al-Kasassbeh, M., & Alnidami, N. (2019). Phishing detection based on machine learning and feature selection methods.

[8] Wen, L., & Yu, H. (2017, August). An Android malware detection system based on machine learning. In AIP

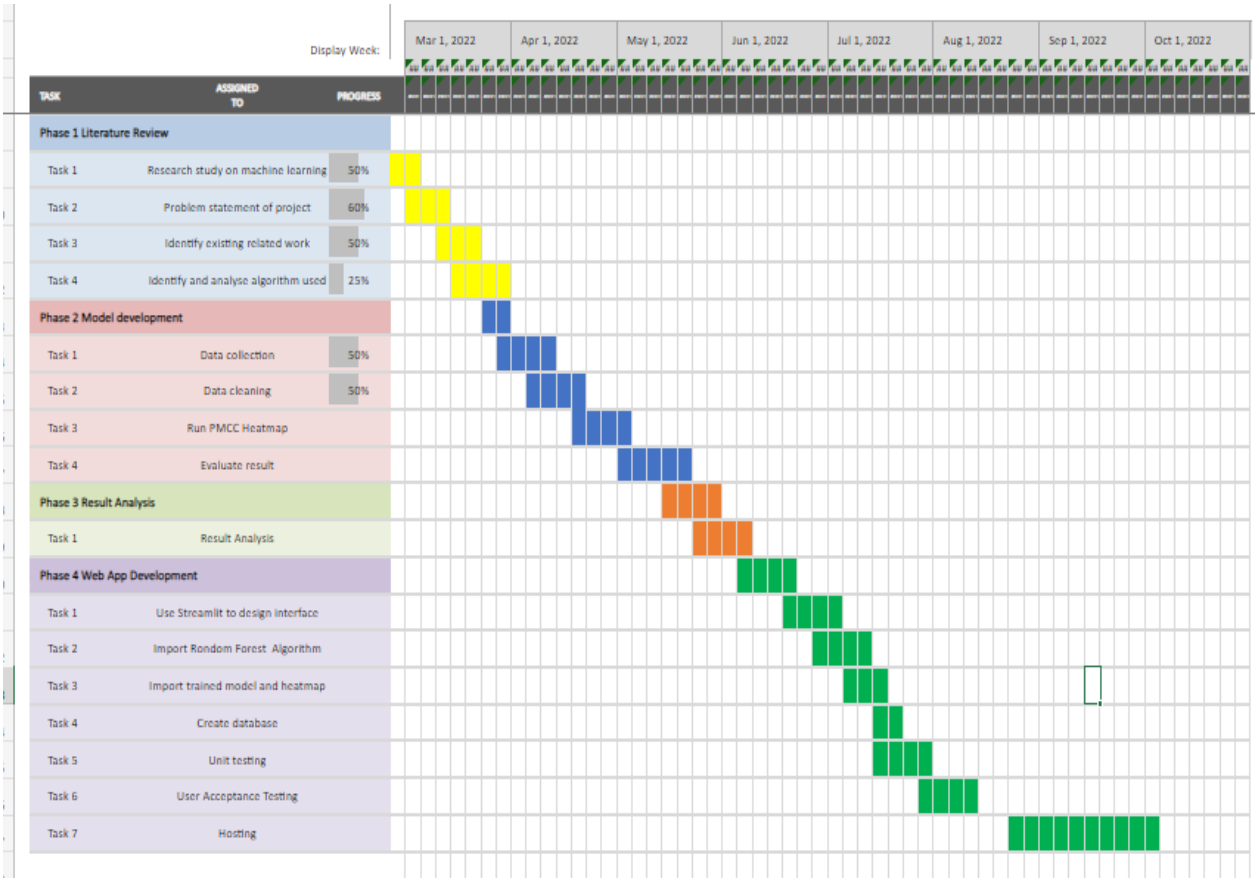
[9] Niklas Donges (2019). A COMPLETE GUIDE TO THE RANDOM FOREST ALGORITHM.
Retrieved from https://builtin.com/data-science/random-forest-algorithm

[10] Great Leaning Team (Feb 19, 2020). Random Forest Algorithm- An Overview.
Retrieved from https://www.mygreatlearning.com/blog/random-forest-algorithm/

[11] ProgrammerSought (n.d.). Understanding of ID3 algorithm and its advantages and disadvantages. Retrieved from https://www.programmersought.com/article/53375975796/

APPENDIX A

GANTT CHART


Display Week:
Mar 1, 2022
Apr 1, 2022
May 1, 2022
Jun 1, 2022
Jul 1, 2022
Aug 1, 2022
Sep 1, 2022
Oct 1, 2022
TASK
ASSIGNED TO
PROGRESS
Phase 1 Literature Review
Task 1 Research study on machine learning 50%
Task 2 Problem statement of project 60%
Task 3 Identify existing related work 50%
Task 4 Identify and analyse algorithm used 25%
Phase 2 Model development
Task 1 Data collection 50%
Task 2 Data cleaning 50%
Task 3 Run PMCC Heatmap
Task 4 Evaluate result
Phase 3 Result Analysis
Task 1 Result Analysis
Phase 4 Web App Development
Task 1 Use Streamlit to design interface
Task 2 Import Rondom Forest Algorithm
Task 3 Import trained model and heatmap
Task 4 Create database
Task 5 Unit testing
Task 6 User Acceptance Testing
Task 7 Hosting

APPENDIX B

SOFTWARE REQUIREMENT SPECIFICATION (SRS)

2022

SOFTWARE REQUIREMENT SPECIFICATION (SRS)

A CRYPTOJACKING DETECTION SYSTEM WITH PRODUCT MOMENT CORRELATION COEFFICIENT (PMCC) HEATMAP INTELLIGENT

KONG JUN HAO [CB19109]
To be submitted to BCC3024 UNDERGRADUATE PROJECT 2
Bachelor of Computer Science (Software Engineering)

DOCUMENT APPROVAL

	Name	Date
Authenticated by: _____ Developer	KONG JUN HAO	
Approved by: _____ Client		

Software :

Archiving Place :

TABLE OF CONTENT

LIST OF FIGURE

LIST OF TABLES

LIST OF APPENDICES

CHAPTER 1

1 INTRODUCTION

1.1 Project description

A cryptojacking detection system with product moment correlation coefficient (PMCC) heatmap intelligent is a system work on detecting cryptojacking threats on the user devices. Before the application is developed, an algorithm and model will be built using Random Forest Classifier in machine learning to train the model until it achieved and produces the most accurate prediction result. The model that is trained successfully is implemented to build the cryptojacking detection system. As such, cryptojacking detection system is a simple application that contain two important components which are input data and view result.

The purpose of (Software Requirement Specification) SRS document is to provide a detailed overview of the components for cryptojacking detection system and the precise implementation details required to satisfy the requirements as specified in the Software Requirements Specification (SRS), its parameters and goals. This document describes how the system is expected to be executed. This document is intended for both the stakeholders and the system developer as a reference to develop the first version of the cryptojacking detection system and defines the system in aspects of its functions, requirements and interfaces.

1.2 System identification

Document Type	: Software Requirement Specification
Document Abbreviations	: SRS
System Title	: A cryptojacking detection system with product moment correlation coefficient (PMCC) heatmap intelligent

System Abbreviations : CDS_PMCC

Establish Year : 2022 (2K22)

Version : 1.0 (V1)

System Identification No. : CDS_PMCC-SRS-2K22-V1

In system identification no., SRS stands for the name of the document. CDS_PMCC stands for the title of the system, A Cryptojacking Detection System with Product Moment Correlation Coefficient (PMCC) Heatmap Intelligent. Next, 2K22, which stands for the year the system is established and V1 stands for the version of the SRS document.

1.3 System overview

A cryptojacking detection system with product moment correlation coefficient (PMCC) heatmap intelligent (CDS_PMCC) is a web-based system used to detect cryptojacking threats on the user devices. The system will be designed to allow all of the stakeholders to input required data into the system and view the cryptojacking prediction result. There are two types of stakeholders involved in this system which are public user and admin. The functionality requirements of CDS_PMCC are input data and view result.

The first features is input data. This function allows users to input the required data into each respective column into the system. The input data are verified and display in the system.

The second features is view result. This function allows users to view the prediction result after users had input the value into the system. The system will display the probability and prediction result to the stakeholders.

1.4 Contex diagram

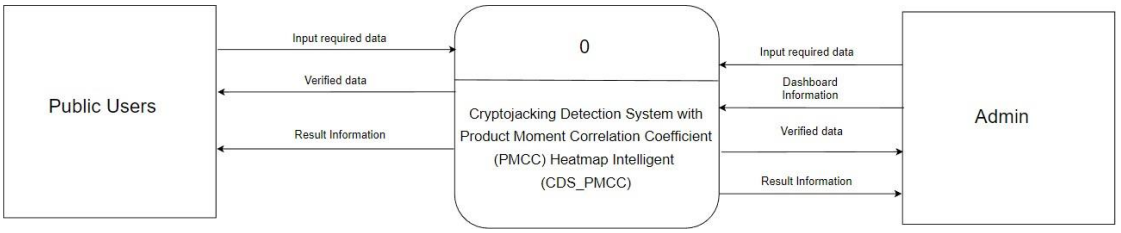


Figure 1.4 Contex diagram for CDS_PMCC.

A cryptojacking detection system with product moment correlation coefficient (PMCC) heatmap intelligent (CDS_PMCC) consists of two entities which are public users and admin. The public users and admin are allow to input the required data into the system. The input data are verified and display in the system. Moreover, the public users and admin are able to view the prediction result after users had input the value into the system. The system will display the probability and prediction result. The admin also able to manage the dashboard.

CHAPTER 2

2 OVERALL DESCRIPTION

2.1 Use case diagram and description

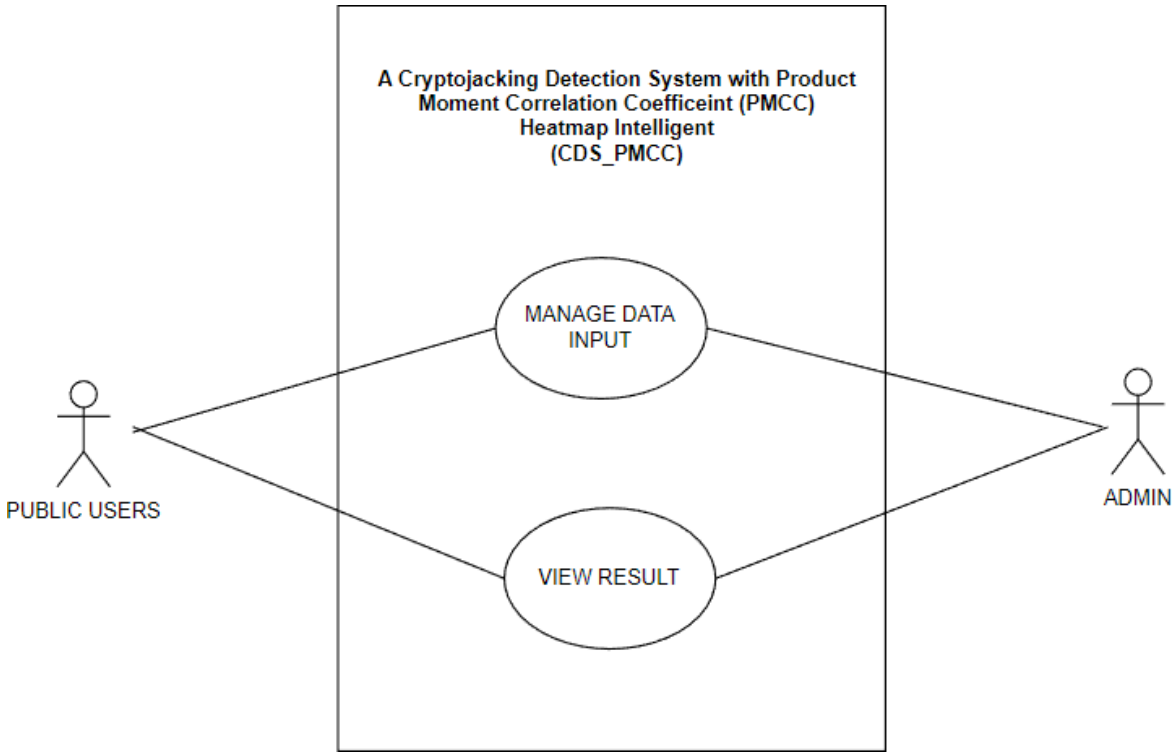


Figure 2.1.1 The overall use case diagram for CDS_PMCC

This system consists of two modules which are manage data input and view result. The table below shows a brief description and actors involved for each module.

Table 2.1 Brief description and actors for each module.

Modules	Description	Actors
Manage data input	This module allows users to input data into the system. The data are verified and display in the system.	Admin and public users.
View Result	This module allows users to view the prediction result after users had input the data.	Admin and public users.

3 DETAIL REQUIREMENTS DESCRIPTION

3.1 Software product features

3.1.1 Manage data input

Use Case Diagram

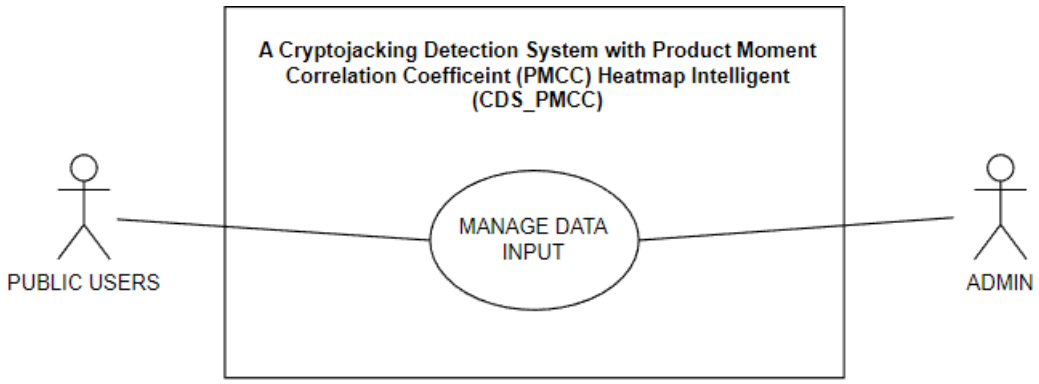


Figure 3.1.1.1 Use case diagram for manage data input

Table 3.1.1 Use case description for manage data input

Use Case Description

Use Case Name	Manage Data Input
Use Case ID	CDS_PMCC_UCI_1000
Brief Description	This use case allows the public users and admin to input data manually

	to the Cryptojacking Detection System.
Actor	Public users and Admin
Pre-conditions	The public users and admin had successfully access to the Cryptojacking Detection System.
Basic flow	1. The use case starts when the users select the user input features. **2.** Fill in data **[CDS_PMCC_UCI_1001]** 3. The public users and admin are required to select the features column. **4.** Users fill in the data in each respective column. **[A1: Invalid data]** 5. The system reads the input data. 6. The system updates the data into the database. 7. The system verifies the data. 8. The system displays the prediction result. 9. The use case end.
Alternative flow	**A1: Invalid data [CDS_PMCC_UCI_1002]** 1. The system display an error message. 2. Users are required fill in the data again with correct format. 3. Use case continue with step 5 in basic flow.
Exception flow	None.

Post-conditions	1. The data input by the users are saved and updated to the system. 2. The system display the prediction results of cryptojacking to the users.
Constraint	None.
Rules	None.
Sequence Diagram	Refer Appendix A
Interface	Refer 3.2.1

3.1.2 View result

Use Case Diagram

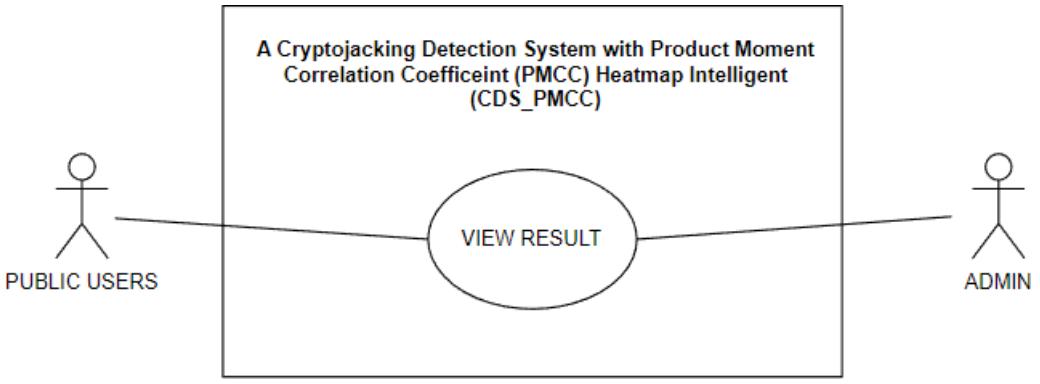


Figure 3.1.2.1 Use case diagram for view result

Table 3.1.2 Use case description for view result

Use Case Description

Use Case Name	View Result
Use Case ID	CDS_PMCC_UCI_2000
Brief Description	This use case allows the public users and admin to view the cryptojacking prediction result of the input data on the Cryptojacking Detection System.
Actor	Public users and Admin.
Pre-conditions	1. The public users and admin successfully access to the Cryptojacking Detection System. 2. The public users and admin had successfully input the data into the system.
Basic flow	1. The use case starts when the public users and admin click on the <<**Result**>> button. **2.** View Result: **[CDS_PMCC_UCI_2001]** 3. The system displays the prediction result of cryptojacking towards the data input by the users. 4. The public users and admin view the prediction results through the prediction probability table and prediction table. 5. The use case end.

Alternative flow	None.
Exception flow	None.
Post-conditions	The system displays the cryptojacking prediction result through the prediction probability table and prediction table and allow users to view the result.
Constraint	None.
Rules	None.
Sequence Diagram	Refer Appendix A
Interface	Refer 3.2.1

3.2 GUI/ Wireframe

3.2.1 Dashboard Interface

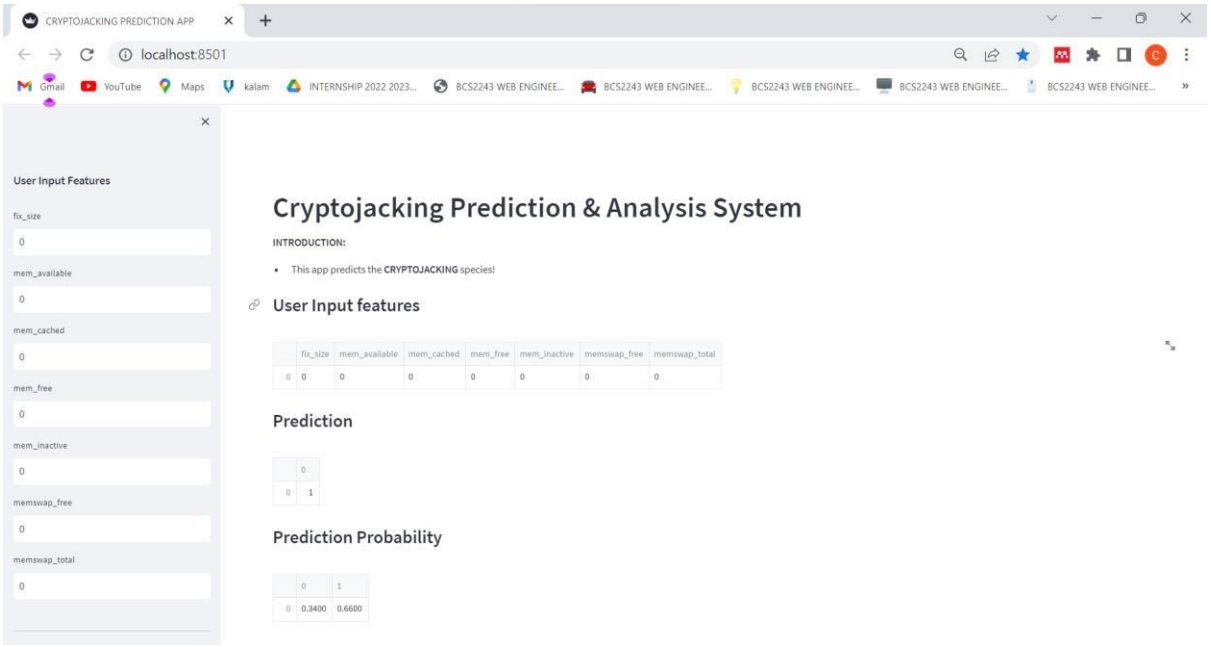


Figure 3.2.1.1 Dashboard Interface of Cryptojacking Detection System (include Manage Data Input and View Result)

3.2.2 Heatmap Interface

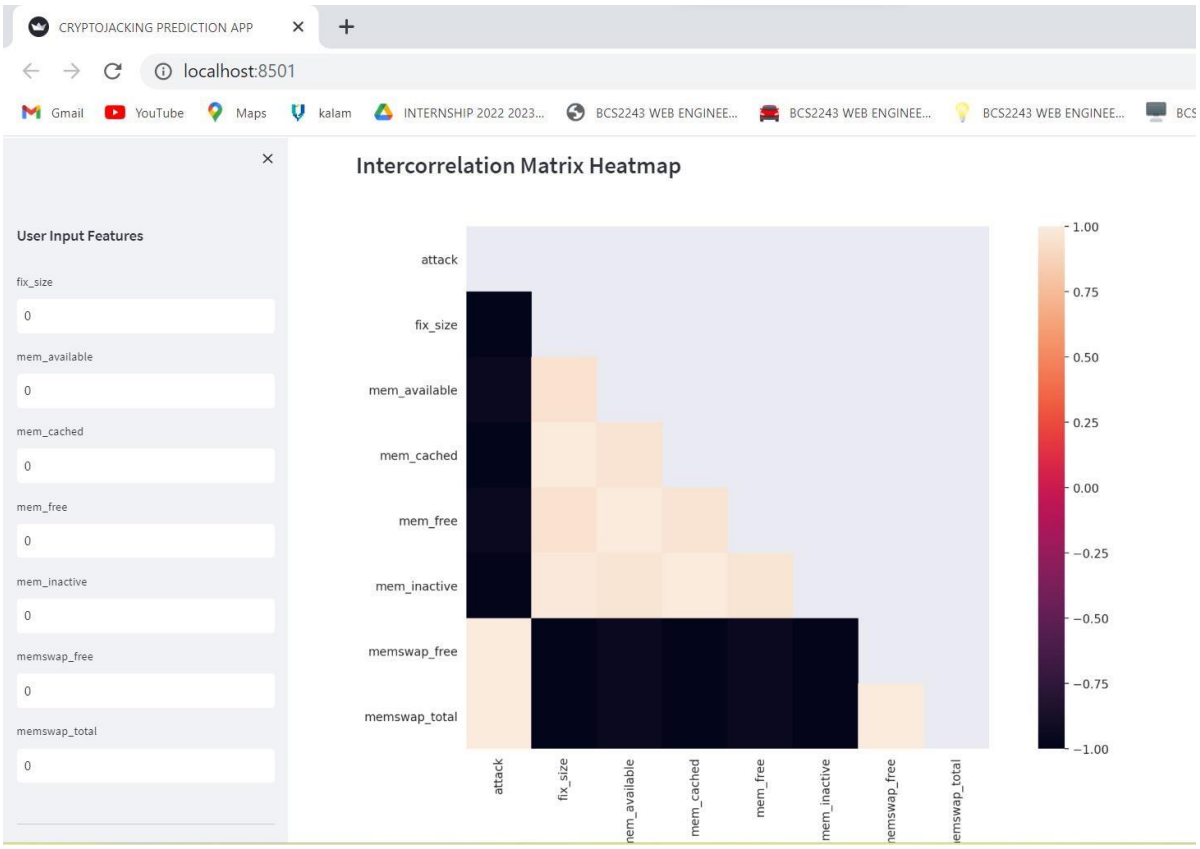


Figure 3.2.2.1 Heatmap Interface of Cryptojacking Detection System

4 HARDWARE AND SOFTWARE SPECIFICATION

Table 4.1 Hardware and software specification.

Name	Version	Type	Description	Purpose
Laptop	LENOVO IdeaPad 3 15ALC6	Hardware	Portable personal computer	Used to develop and run the software.
Python	3.9	Software	Python is an explanatory, objectoriented, high-level programming language with dynamic semantics.	Used to edit and develop the python language to create the algorithm.
Jupyter Notebook	6.3.0	Software	Jupyter notebook is a web-based interactive development environment for notebooks, code, and data.	Used to test and develop the system. This software can help to get the result while running the system.

Microsoft Office	2021	Software	Microsoft Office is a set of applications designed to help improve work efficiency and complete common tasks on your computer.	Used for drawing ,complete thereport in this project.
Google Chrome	100.0.4896.88	Software	Google Chrome browser is an open source program for accessing the World Wide Web and running Web-based applications.	Used to search related information and references for the project.
GitHub	2.9.0	Software	GitHub is a code hosting platform for version control and collaboration. It allow user to work together from anywhere.	Used to upload and store coding of the project.

5 APPENDIX A: SEQUENCE DIAGRAM

Refer Use Case ID: CDS_PMCC_UCI_1000

5.1 Sequence diagram for manage data input

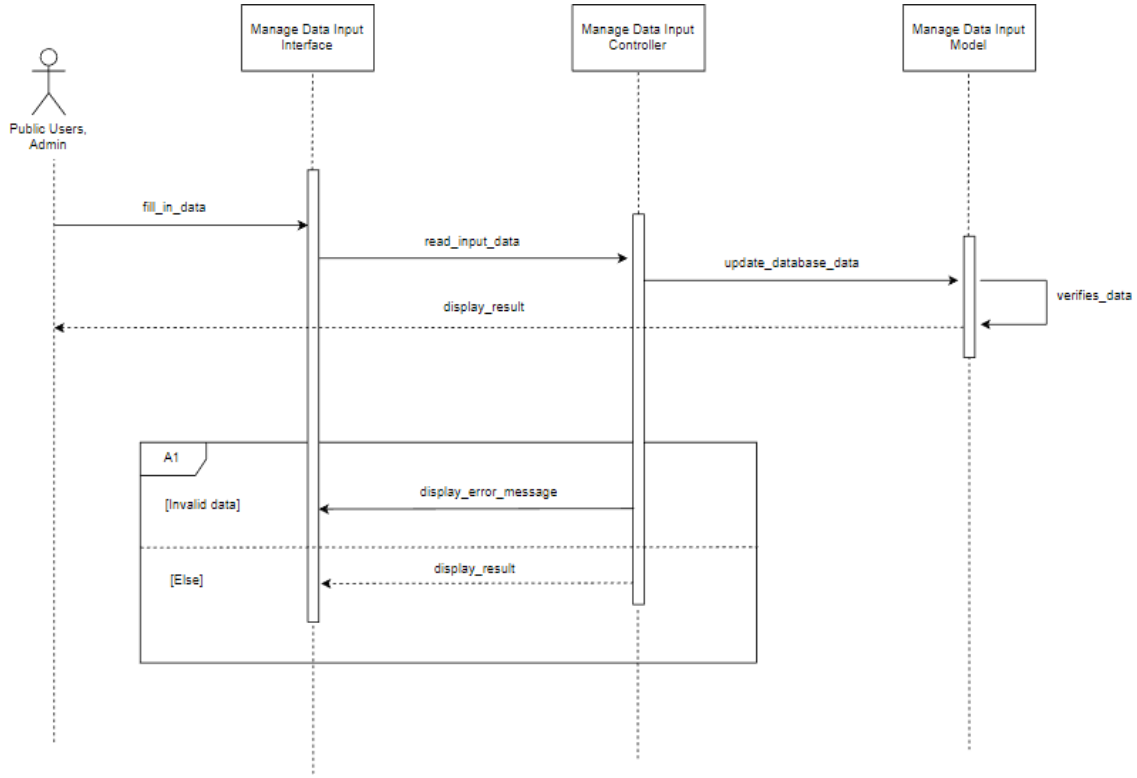


Figure 5.1.1 Sequence diagram for manage data input.

Refer Use Case ID: CDS_PMCC_UCI_2000

5.2 Sequence diagram for view result

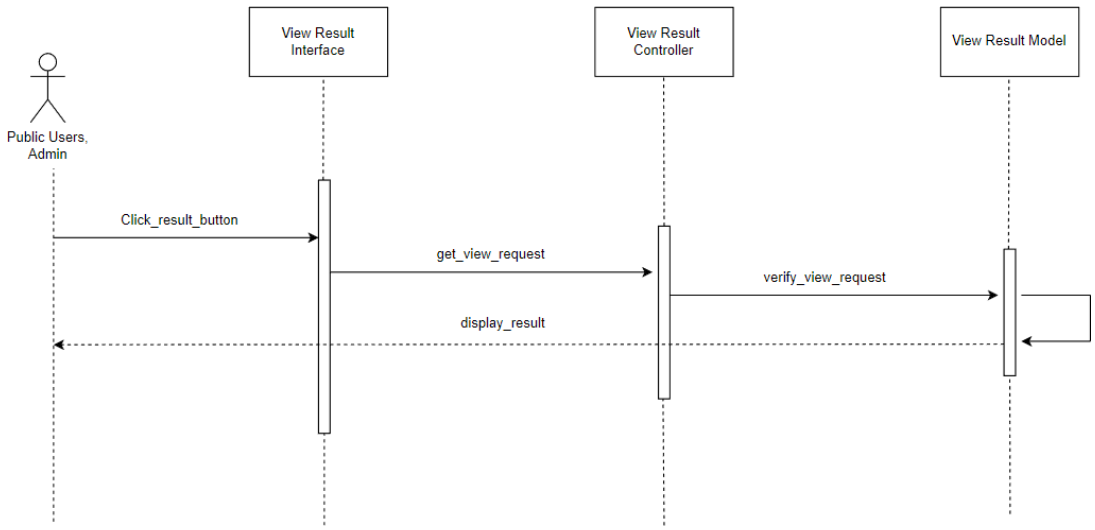


Figure 5.2.1 Sequence diagram for view result

6 APPENDIX B: ACTIVITY DIAGRAM

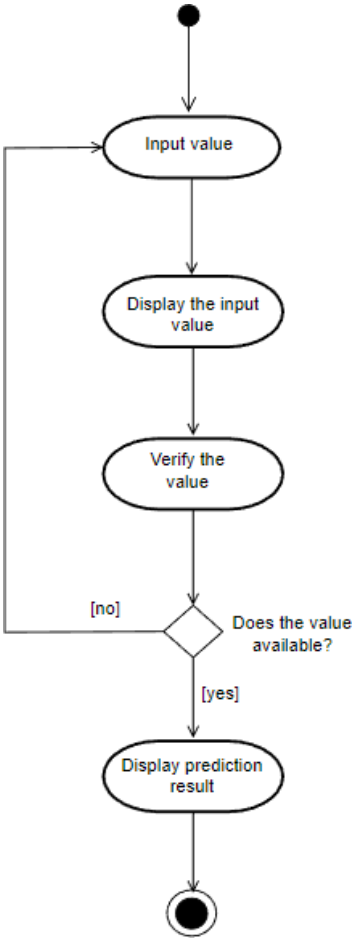


Figure 6.1 Activity diagram of Cryptojacking detection System

7 APPENDIX C:ACRONYMS and ABBREVIATION

Table 7.1 Acronyms and Abbreviation

Term	**Definition**
V1	Version 1
GUI	Graphic User Interface
ID	Identification Number
SRS	Software Requirement Specification

CDS	Cryptojacking Detection System
PMCC	Product Moment Correlation Coefficient (PMCC)

8 APPENDIX D: TRACEABILITY MATRIX

Table 8.1 Traceability Matrix

Related Use Case ID	Requirement ID	Requirement Statement
CDS_PMCC_UCI_1000	CDS_PMCC_UCI_1001	The public users and admin are required to fill in the data in to the system.
	CDS_PMCC_UCI_1002	A1: Invalid data The data inserted are not valid.
CDS_PMCC_UCI_2000	CDS_PMCC_UCI_2001	The public users and admin are allow to view the prediction result.

APPENDIX C

SOFTWARE DESIGN DOCUMENT (SDD)

2022

SOFTWARE DESIGN DESCRIPTION (SDD)

A CRYPTOJACKING DETECTION SYSTEM WITH PRODUCT MOMENT CORRELATION COEFFICIENT (PMCC) HEATMAP INTELLIGENT

KONG JUN HAO [CB19109]
To be submitted to BCC3024 UNDERGRADUATE PROJECT 2
Bachelor of Computer Science (Software Engineering)

DOCUMENT APPROVAL

	Name	Date
Authenticated by: _____ Name:	KONG JUN HAO	
Approved by: _____ Client		

Software :

Archiving Place :

TABLE OF CONTENT

Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

CHAPTER 1

9 INTRODUCTION

9.1 PROJECT DESCRIPTION

A cryptojacking detection system with product moment correlation coefficient (PMCC) heatmap intelligent is a system work on detecting cryptojacking threats on the user devices. Before the application is developed, an algorithm and model will be built using Random Forest Classifier in machine learning to train the model until it achieved and produces the most accurate prediction result. The model that is trained successfully is implemented to build the cryptojacking detection system. As such, cryptojacking detection system is a simple application that contain two important components which are input data and view result.

The purpose of (Software Design Document) SDD is to provide a detailed architecture and design of the components for cryptojacking detection system and the precise implementation details required to satisfy the requirements as specified in the Software Design Document (SDD), its parameters and goals. This document describes the architectures used to build the system. This document is intended for both the stakeholders and the system developer as a reference to develop the first version of the cryptojacking detection system and defines the system in aspects of its functions, requirements and interfacesg.

9.2 SYSTEM IDENTIFICATION

Document Type	: Software Design Document
Document Abbreviations	: SDD
System Title	: A cryptojacking detection system with product moment correlation coefficient (PMCC) heatmap intelligent
System Abbreviations	: CDS_PMCC
Establish Year	: 2022 (2K22)

Version : 1.0 (V1)

Identification No. : CDS_PMCC-SDD-2K22-V1

In identification no., SDD stands for the name of the document. CDS_PMCC stands for the title of the system, A Cryptojacking Detection System with Product Moment Correlation Coefficient (PMCC) Heatmap Intelligent. Next, 2K22, which stands for the year the system is established and V1 stands for the version of the SDD document.

9.3 SYSTEM OVERVIEW

A cryptojacking detection system with product moment correlation coefficient (PMCC) heatmap intelligent (CDS_PMCC) is a web-based system used to detect cryptojacking threats on the user devices. The system will be designed to allow all of the stakeholders to input required data into the system and view the cryptojacking prediction result. The stakeholders that involve in this system are the public users. The functionality requirements of CDS_PMCC are input data and view result.

The first features is input data. This function allows users to input the required data into each respective column into the system. The input data are verified and display in the system.

The second features is view result. This function allows users to view the prediction result after users had input the value into the system. The system will display the probability and prediction result to the stakeholders.

10 GENERAL ARCHITECTURE

10.1 APPLICATION LAYER

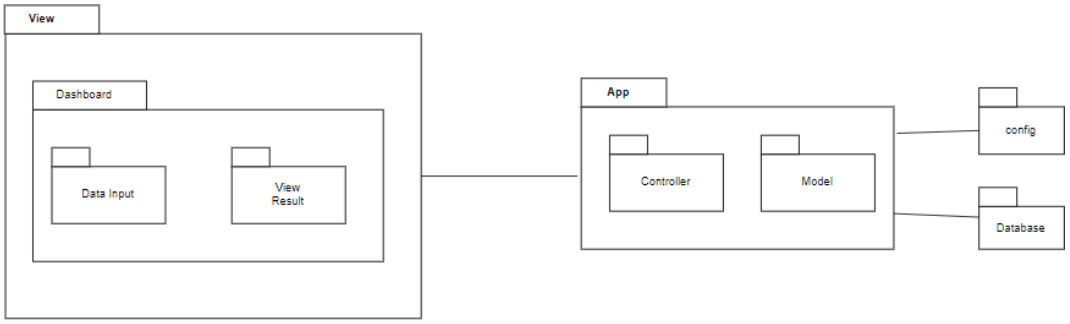


Figure 2.1.1 General Architecture.

10.1.1 MANAGE DATA INPUT [SDD-REQ-1000]

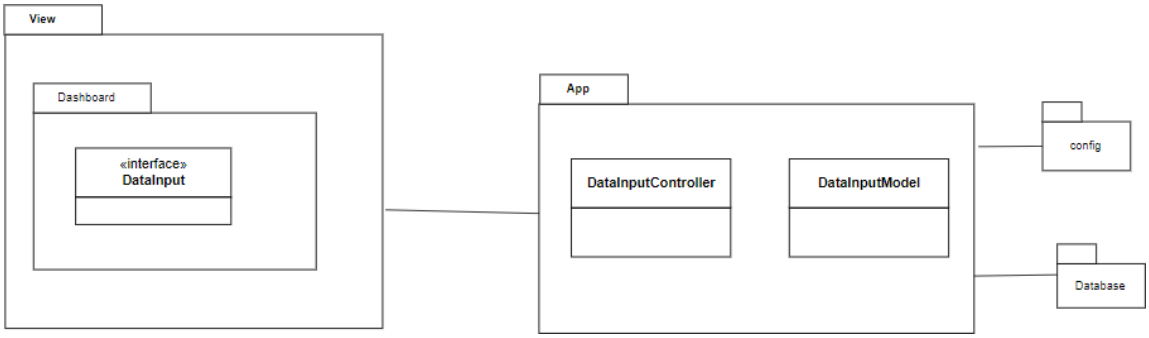


Figure 2.1.1.1 Manage Data Input Application Layer.

10.1.1.1 Manage Data Input View

Table 2.1.1.1 Manage Data Input View.

Class Name	Description
DataInput	This interface allows public users to insert the data into the system.

10.1.1.2 Manage Data Input Application

Table 2.1.1.2 Manage Data Input Application.

Class Name	Description
DataInputController	The controller is used to manage the data between the interface and model.
DataInputModel	This model is used to retrieve the data in the database.

10.1.2 VIEW RESULT [SDD-REQ-2000]

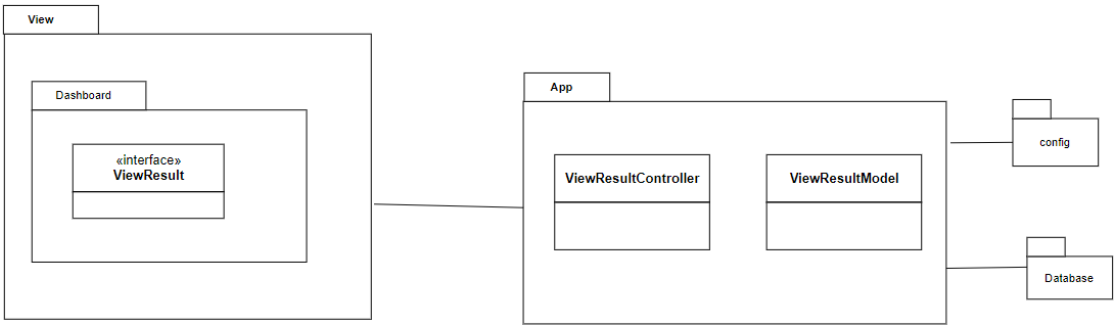


Figure 2.1.2.1 Manage View Result Application Layer.

10.1.2.1 Manage View Result View

Table 2.1.2.1 Manage View Result View.

Class Name	Description
ViewResult	This interface allows public users to view the prediction result and prediction probability.

10.1.2.2 Manage View Result Application

Table 2.1.2.2 Manage View Result View application

Class Name	Description
ViewResultController	The controller is used to manage the result display between the interface and model.
ViewResultModel	This model is used to retrieve the data in the database.

10.2 MIDDLEWARE LAYER

Middleware handles connections between application software and the underlying layers of system software, such as the operating system and the device driver layer. There are several different software and elements that CDS_PMCC will used throughout the development phase.

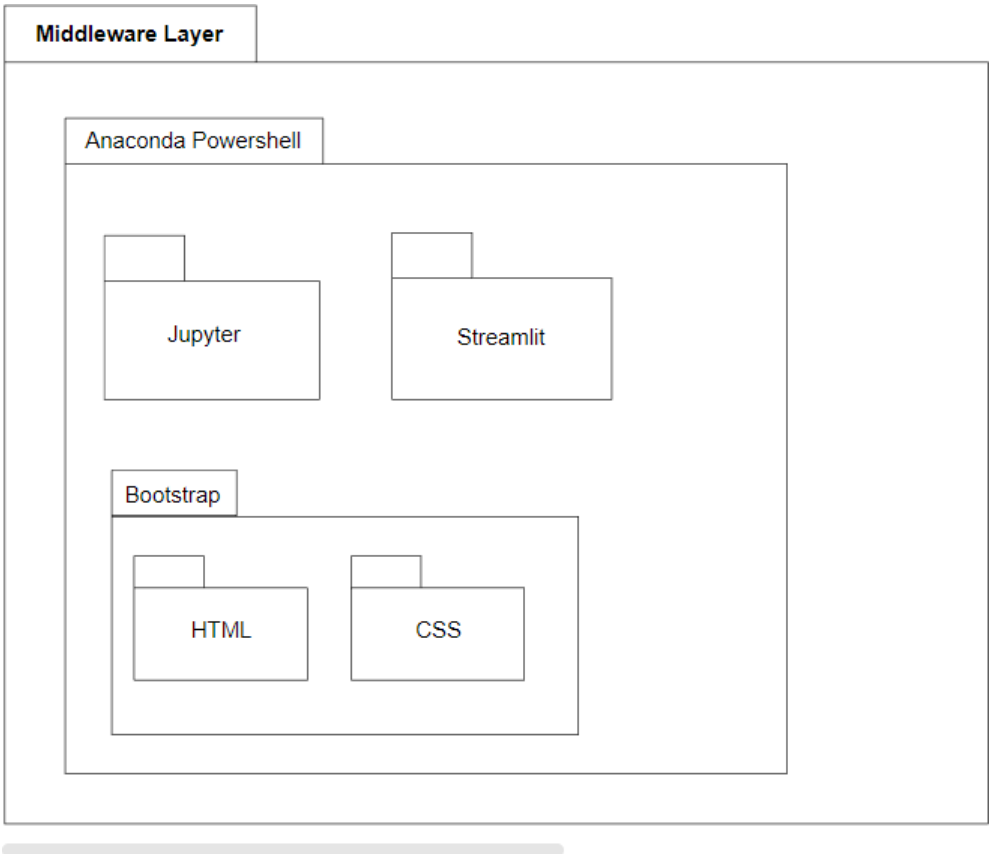


Figure 2.2.1 Middleware Layer of CDS_PMCC

Table 2.2 Middleware Layer Description

Package Name	Description
Jupyter	Jupyter is used to train and generate a visualization of the heatmap.
Streamlit	Streamlit provide a web-framework to develop the dashboard of the application.
Bootstrap	Bootstrap is the programming language used to design the application.
HTML	HTML is used to structured the web page.
CSS	CSS is used to describe how HTML elements are to be displayed to users.
Ananconda Powershell	Ananconda Powershell provides an environment to run Python.

CHAPTER 2

2.1 DETAILED DESCRIPTION

10.2.1 Package 1: Manage Data Input [SDD-REQ-1000]

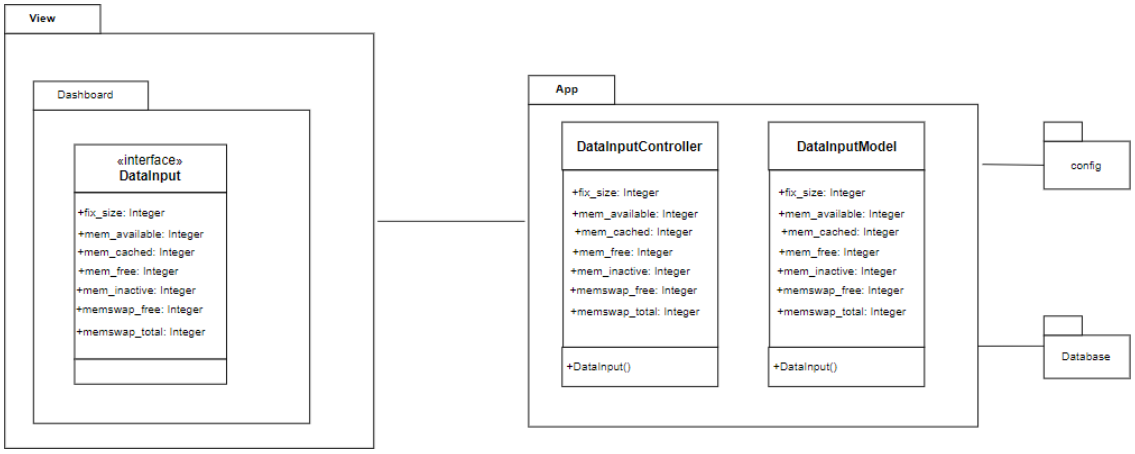


Figure 2.2.1.1 Manage Data Input package

10.2.1.1 DataInput [SDD-REQ-1001]

Table 2.2.1.1 DataInput boundary class.

Class Type	Boundary Class	
Responsibility	An interface that allows public users to input the required data into the system.	
Attributes	**Attributes Name**	**Attributes Type**
	fix_size	Integer
	mem_available	Integer
	mem_cached	Integer
	mem_free	Integer

	mem_inactive memswap_free memswap_total	Integer Integer Integer
Methods	**Method Name**	**Description**
	Not applicable	Not applicable
Algorithm	BEGIN System displays the dashboard interface. Users enter fix_size Users enter mem_available Users enter mem_cached Users enter mem_free Users enter mem_inactive Users enter memswap_free Users enter memswap_total The data input by the user will be display in the system END	

10.2.1.2 DataInputController [SDD_REQ_1002]

Table 2.2.1.2 DataInputController controller class.

Class Type	Controller Class	
Responsibility	A controller that control the flow of input and output between interface of DataInput and entity class DataInputModel.	
Attributes	**Attributes Name**	**Attributes Type**

	fix_size mem_available mem_cached mem_free mem_inactive memswap_free memswap_total	Integer Integer Integer Integer Integer Integer Integer
Methods	**Method Name**	**Description**
	DataInput()	This method is used for public users to input data into the system.

Algorithm	**DataInput()**
	BEGIN
	ADD fix_size
	IF fix_size=fix_size
	DISPLAY in the DataInput interface
	ELSE display error message
	END IF
	ADD mem_available
	IF mem_available=mem_available
	DISPLAY in the DataInput interface
	ELSE display error message
	END IF
	ADD mem_cached
	IF mem_cached=mem_cached	

	DISPLAY in the DataInput interface ELSE display error message END IF ADD mem_free IF mem_free=mem_free DISPLAY in the DataInput interface ELSE display error message END IF ADD mem_inactive IF mem_inactive=mem_inactive DISPLAY in the DataInput interface ELSE display error message END IF ADD memswap_free IF memswap_free=memswap_free DISPLAY in the DataInput interface ELSE display error message END IF ADD memswap_total IF memswap_total=memswap_total DISPLAY in the DataInput interface ELSE display error message END IF END

10.2.1.3 DataInputModel [SDD_REQ-1003]

Table 2.2.1.3 DataInputModel entity class.

Class Type	Entity class	
Responsibility	An entity class that responsible to retrieve data from DataInputController and verify the data in the database	
Attributes	**Attributes Name**	**Attributes Type**
	fix_size mem_available mem_cached mem_free mem_inactive memswap_free memswap_total	Integer Integer Integer Integer Integer Integer Integer
Methods	**Method Name**	**Description**
	DataInput()	This method is used to retrieve data from the DataInputController and send to database

Algorithm	**DataInput()**
	BEGIN
	READ fix_size
	IF fix_size=fix_size
	DISPLAY in the DataInput interface
	ELSE display error message
	END IF
	READ mem_available	

	IF mem_available=mem_available DISPLAY in the DataInput interface ELSE display error message END IF READ mem_cached IF mem_cached=mem_cached DISPLAY in the DataInput interface ELSE display error message END IF READ mem_free IF mem_free=mem_free DISPLAY in the DataInput interface ELSE display error message END IF READ mem_inactive IF mem_inactive=mem_inactive DISPLAY in the DataInput interface ELSE display error message END IF READ memswap_free IF memswap_free=memswap_free DISPLAY in the DataInput interface ELSE display error message END IF READ memswap_total IF memswap_total=memswap_total

	DISPLAY in the DataInput interface ELSE display error message END IF END

10.2.2 Package 2: View Result [SDD-REQ-2000]

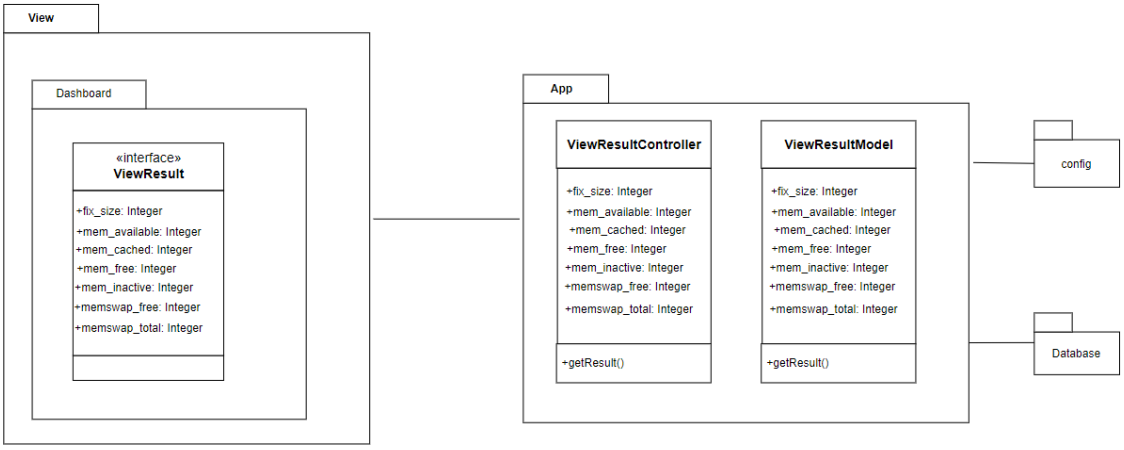


Figure 2.2.2.1 View Result package

10.2.2.1 ViewResult [SDD-REQ-2001]

Table 2.2.2.1 ViewResult boundary class

Class Type	Boundary class	
Responsibility	To display and allow users to view the prediction result.	
Attributes	**Attributes Name**	**Attributes Type**

	fix_size mem_available mem_cached mem_free mem_inactive memswap_free memswap_total	Integer Integer Integer Integer Integer Integer Integer
Methods	**Method Name**	**Description**
	Not applicable	Not applicable
Algorithm	BEGIN System displays the Dashboard interface DISPLAY fix_size DISPLAY mem_available DISPLAY mem_cached DISPLAY mem_free DISPLAY mem_inactive DISPLAY memswap_free DISPLAY memswap_total System displays the prediction result END	

10.2.2.2 ViewResultController [SDD-REQ-2002]

Table 2.2.2.2 ViewResultController controller class

<table>
<tr><td>Class Type</td><td colspan="2">Controller class</td></tr>
<tr><td>Responsibility</td><td colspan="2">A controller used to control the flow of input and output between interface of ViewResult and entity class ViewResultModel.</td></tr>
<tr><td rowspan="2">Attributes</td><td>Attributes Name</td><td>Attributes Type</td></tr>
<tr><td>fix_size
mem_available
mem_cached
mem_free
mem_inactive
memswap_free
memswap_total</td><td>Integer
Integer
Integer
Integer
Integer
Integer
Integer</td></tr>
<tr><td rowspan="2">Methods</td><td>Method Name</td><td>Description</td></tr>
<tr><td>getResult ()</td><td>To get result from the ViewResultModel entity class and display on the ViewResult boundary class.</td></tr>
<tr><td>Algorithm</td><td colspan="2">getResult()
BEGIN
READ fix_size
IF fix_size=fix_size
DISPLAY in the ViewResult interface
ELSE display error message
END IF
READ mem_available
IF mem_available=mem_available
DISPLAY in the ViewResult interface
ELSE display error message</td></tr>
</table>

	END IF READ mem_cached IF mem_cached=mem_cached DISPLAY in the ViewResult interface ELSE display error message END IF READ mem_free IF mem_free=mem_free DISPLAY in the ViewResult interface ELSE display error message END IF READ mem_inactive IF mem_inactive=mem_inactive DISPLAY in the ViewResult interface ELSE display error message END IF READ memswap_free IF memswap_free=memswap_free DISPLAY in the ViewResult interface ELSE display error message END IF READ memswap_total IF memswap_total=memswap_total DISPLAY in the ViewResult interface ELSE display error message END IF

	END

10.2.2.3 ViewResultModel [SDD-REQ-2003]

Table 2.2.2.3 ViewResultModel entity class

Class Type	Entity class	
Responsibility	An entity class that responsible to send and retrieve data from ViewResultController and verify the data in the database	
Attributes	**Attributes Name**	**Attributes Type**
	fix_size	Integer
	mem_available	Integer
	mem_cached	Integer
	mem_free	Integer
	mem_inactive	Integer
	memswap_free	Integer
	memswap_total	Integer
Methods	**Method Name**	**Description**
	getResult()	This method is used to retrieve data from the ViewresultController and send to database
Algorithm	**getResult()** BEGIN READ fix_size	

	IF fix_size=fix_size System retrieves the data from database System send all data to ViewResultController class ELSE display error message END IF READ mem_available IF mem_available=mem_available System retrieves the data from database System send all data to ViewResultController class ELSE display error message END IF READ mem_cached IF mem_cached=mem_cached System retrieves the data from database System send all data to ViewResultController class ELSE display error message END IF READ mem_free IF mem_free=mem_free System retrieves the data from database System send all data to ViewResultController class ELSE display error message END IF READ mem_inactive IF mem_inactive=mem_inactive System retrieves the data from database

	System send all data to ViewResultController class ELSE display error message END IF READ memswap_free IF memswap_free=memswap_free System retrieves the data from database System send all data to ViewResultController class ELSE display error message END IF READ memswap_total IF memswap_total=memswap_total System retrieves the data from database System send all data to ViewResultController class ELSE display error message END IF END

10.3 DATA DICTIONARY

Data dictionary provides information such as attributes name, description of attributes, data type of attributes, primary key (PK) and foreign key (FK). Below shows all the data dictionary listed in tables for CDS_PMCC.

10.3.1 Public user

Table 2.3.1 Data Dictionary for table public user

FIELD NAME	DATA TYPE	DESCRIPTION	CONSTRAINT
userID	VARCHAR(255)	User identification	PK
username	VARCHAR(255)	The name of the user account	
userpass	VARCHAR(255)	The password of the user account	

CHAPTER 3

11 TRACEABILITY

11.1 REQUIREMENT TRACEABILITY

Table 3.1 Requirement traceability for CDS_PMCC.

SRS Use Case ID	Description	Design ID
CDS_PMCC_UCI_1000	Allow users to input data into the system.	SDD-REQ-1000 SDD-REQ-1001 SDD-REQ-1002 SDD-REQ-1003
CDS_PMCC_UCI_2000	Allow users to view the prediction result.	SDD-REQ-2000 SDD-REQ-2001 SDD-REQ-2002 SDD-REQ-2003

APPENDIX D

SOFTWARE TESTING DOCUMENT

(STD)

2022

SOFTWARE TESTING DESCRIPTION (STD)

A CRYPTOJACKING DETECTION SYSTEM WITH PRODUCT MOMENT CORRELATION COEFFICIENT (PMCC) HEATMAP INTELLIGENT

KONG JUN HAO [CB19109]
To be submitted to BCC3024 UNDERGRADUATE PROJECT 2
Bachelor of Computer Science (Software Engineering)

DOCUMENT APPROVAL

	Name	Date
Authenticated by: _____ Developer	KONG JUN HAO	
Approved by: _____ Client		

Software :

Archiving Place :

TABLE OF CONTENT

LIST OF FIGURE

LIST OF TABLES

LIST OF APPENDICES

CHAPTER 1

12 INTRODUCTION

1.1 PURPOSE OF DOCUMENT

This software test description supports the following objectives:

i. To identify the test items / modules that will be covered.

ii. To develop the test case for unit testing.

iii. To record the test result from the unit testing.

1.2 SYSTEM IDENTIFICATION

Document Type	: Software Requirement Specification
Document Abbreviations	: SRS
System Title	: A cryptojacking detection system with product moment correlation coefficient (PMCC) heatmap intelligent
System Abbreviations	: CDS_PMCC
Establish Year	: 2022 (2K22)
Version	: 1.0 (V1)
System Identification No.	: CDS_PMCC-SRS-2K22-V1

In system identification no., SRS stands for the name of the document. CDS_PMCC stands for the title of the system, A Cryptojacking Detection System with Product Moment

Correlation Coefficient (PMCC) Heatmap Intelligent. Next, 2K22, which stands for the year the system is established and V1 stands for the version of the SRS document.

1.3 SYSTEM OVERVIEW

A cryptojacking detection system with product moment correlation coefficient (PMCC) heatmap intelligent (CDS_PMCC) is a web-based system used to detect cryptojacking threats on the user devices. The system will be designed to allow all of the stakeholders to input the required data into the system and view the cryptojacking prediction result. The system involves two types of stakeholders: the public users and the admin. The functionality requirements of CDS_PMCC are data input and view result.

The first features is input data. This function allows users to input the required data into each respective column into the system. The input data are verified and display in the system.

The second features is view result. This function allows users to view the prediction result after users had input the value into the system. The system will display the probability and prediction result to the stakeholders.

CHAPTER 2

13 TEST CASE OF THE SYSTEM

2.1 TEST CASE NAME

Table 2.1 Test case description

Test Case ID	Test Case Name
TC01	Manage Data Input
TC02	View Result

2.2 HARDWARE

Table 2.2 Hardware for testing

Name	Description
IdeaPad 3 15ALC6	Laptop used for documentation and testing.

2.3 SOFTWARE

Table 2.3 Software for testing

Name	Description
Streamlit	Used to build web application.
Visual Studio Code	A streamlined code editor with support for development operations like debugging, task running, and version control.
GitHub	To ship better code through command line features by hosting repositories and branches. Make the actions of

	code review and implementation go smooth.

2.4 TESTING SCHEDULE

Table 2.4: Testing Schedule

Test Case ID	Test Name	Date Start	Date End
TC01	Manage Data Input	10/12/2022	17/12/2022
TC02	View Result	18/12/2022	24/12/2022

CHAPTER 3

14 TEST CASE DETAILED DESCRIPTION

3.1 MANAGE DATA INPUT

Table 3.1: Manage data input test case ID and description.

Test case ID	Test Case Description
TC01-01	Public user and admin are able to insert data into the system.

Table 3.2: Test Case to add data to the system.

Test Case ID	TC01-01
Objective	Public user and admin are able to insert data into the system.
Description of Test	Public user and admin are able to input the required data into the system.
Test Input	1. Tester input the data into the respective column in the dashaboard interface, fix_size = 6745954 mem_available = 756456 mem_cached = 4574855 mem_free = 8673454 mem_inactive = -873445 memswap_free = -776465 memswap_total = 674958
Expected Result	1. The interface displays all the data inserted, fix_size = 6745954 mem_available = 756456 mem_cached = 4574855

	mem_free = 8673454 mem_inactive = -873445 memswap_free = -776465 memswap_total = 674958
Criteria for Evaluating Result	1. The data inserted by the users are available and displayed in the interface.
Test Procedure	1. Tester opens the dashboard interface. 2. Tester opens the "User Input Features". 3. Tester input the data into the respective column. 4. The data inserted are available and displayed in the dashboard interface.
Assumption and Constraint	N/A

3.2 VIEW RESULT

Table 3.2: View result test case ID and description.

Test case ID	Test Case Description
TC02-01	Public user and admin are able to view the result.

Table 3.21: Test case to view result.

Test Case ID	TC02-01
Objective	Public user and admin are able to view the result.
Description of Test	Public user and admin are able view the prediction result display by the system.

Test Input	1. Tester input the data into the respective column in the dashboard interface, fix_size = 6745954 mem_available = 756456 mem_cached = 4574855 mem_free = 8673454 mem_inactive = -873445 memswap_free = -776465 memswap_total = 674958
Expected Result	1. The system displays the prediction result and prediction probability.
Criteria for Evaluating Result	1. The prediction result and prediction probability are correct and displayed.
Test Procedure	2. Tester opens the dashboard interface. 3. Tester opens the "User Input Features". 4. Tester input the data into the respective column. 5. The system displays the prediction result and prediction probability. 6. The tester compares the displayed data and the data from the database.
Assumption and Constraint	N/A

CHAPTER 4

15 TEST RESULT

4.1 TEST RESULT OF ALL TEST CASES

Table 4.1: The test result of all test cases.

Test Case	Test Case ID	Status	Failure	Remarks
Public user and admin are able to insert data into the system.	TC01-01	COMPLETED	NO	N/A
Public user and admin are able to view the result.	TC02-01	COMPLETED	NO	N/A