National College of Ireland

# Cryptojacking detection using CPU Utilization as a target attribute with machine learning techniques

MSc Research Project
MSc Cyber Security

Snehal Sarjerao Bhosale
Student ID: x19213948

School of Computing
National College of Ireland

Supervisor: Imran Khan

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

| | |
|---|---|
| **Student Name:** | Snehal Sarjerao Bhosale |
| **Student ID:** | X19213948 |
| **Programme:** | MSc in Cybersecurity **Year:** 2021-2022 |
| **Module:** | Research Project |
| **Supervisor:** | Imran Khan |
| **Submission Due Date:** | 26/04/2022 |
| **Project Title:** | Cryptojacking detection using CPU utilization as a target attribute with machine learning techniques |
| **Word Count:** | **6438 Page Count 27** |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** Snehal Sarjerao Bhosale

**Date:** 26/04/2022

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies) | □ |
| **Attach a Moodle submission receipt of the online project submission,** to each project (including multiple copies). | □ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | □ |

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Cryptojacking detection using CPU Utilization as a target attribute with machine learning techniques

Snehal Sarjerao Bhosale
X19213948

**Abstract**

The new cybersecurity attack, in which the enemy illegally uses crypto-mining software on devices users are unaware of, is known as cryptojacking which proved to be very effective in view of the ease of use of the crypto-client device. A few resistance measures have previously introduced, with distinctive capabilities and functionality, although all are signalized by a host-based structure. These sorts of services, established to guard each user, are not meant to effectively protect the business network.

Malicious hackers are presently using cryptojacking to their advantage. This sort of virus infiltrates users' machines despite their knowledge. It frequently attacks websites and uses complex CPU computations to generate bitcoins in the account of a computer hacker who corporates without accounting for the energy required. This sort of violence degrades system productivity and potentially impair the equipment's lifespan. A revolutionary method for detecting cryptojacking has been proposed, which involves tracking the CPU utilization of accessed internet sites. The research was successful in achieving measures like accuracy and precision close to 1 by incorporating a range of CPU measuring characteristics with the deployment of a scanning device.

This report proposed a Machine Learning (ML)-based framework aiming at finding activities related to cryptocurrencies. In view of the magnitude and severity of the prepared threat it is believed that the concept, substantiated by impressive gains, will pave the ground for more study in this domain.

**Keywords: Cryptojacking, Cryptomining, CPU monitoring, Decision tree, Random Forest etc.**

## 1. Introduction

Designing and Utilizing Web Cryptomining Discovery Learning Techniques" Over the years, cryptocurrencies such as Bitcoin, Monero and Ethereum have gained popularity as they provide an effective alternative to the central banking system and a profitable context for financial speculation. A key component of the cryptocurrency structure is the mining process, in which a complex computer cryptographic problem has to be solved in order to secure a group of online operations and generate a new currency. As this machine creates a reward for each problem solved correctly, some malicious users, instead of using their own tools, began to make website visitors use silent cryptomining code on their machines, which is actually a new source of profit.

The process, aimed at exploiting third-party device resources, has been termed 'cryptojacking' or 'drive-by mining': it contains a new web threat aimed at secretly hiding users who steal money to capture cryptocurrency while browsing the infected person. Website; as reported by most security providers at the time (2017-2018), cryptohighjacking attacks have become more widespread, which at risk are hitting and causing annoying problems for users using the internet. Initially, web-based mines were intended to be used by websites as a new model to make money instead of ads, but it soon became apparent that criminals were being exploited to make botnets of active devices to profit from the victim.

Cybercriminals can infect websites by inserting malicious JavaScript code into their source page: using newly developed code libraries, which are mainly owned by third-party domains, those websites start a secret mining process every time a user logs into a malicious web page. In addition, to make such a method faster and more efficient, technologies such as WebWorkers and WebAssembly are widely used even though they are designed for other purposes. Today, most websites and web applications rely on JavaScript to work properly so it is very important to be able to isolate and disable it. It should be emphasized that, although today the threat of crypto theft is widely known by many companies promoting IT security, it is no longer a matter of designing and implementing an independent detection system, which can effectively protect users.

The intention of this investigation is to examine in depth the risks of crypto theft and to develop a way to automatically find malicious scripts on cryptojacking websites, which provides users with better and more secure filtering information. The text acquisition algorithm designed is based on the dynamic analysis of JavaScript APIs used during operation; a set of calling tasks and related events are provided by the separator as predictive features and labeled text as cryptojacker or not.

The flexible approach presents many benefits such as being able to work even when harmful text is blurred, but it also has many limitations such as the lack of offline analysis. The operating system has shown good results, revealing a flexible and efficient way to detect cryptojacker operations on the web.

## 1.1. Background Scope

Cryptocurrency mining ensures a variety of cryptocurrency transactions and blockchains are added to the blockchain. This is a very important factor in maintaining the sequence of action active. In the procedure, Minors are offered an inducement in the manner of digital asset certification. The profit a miner makes when using their system is not significant over the entire investment period (Bonneau, et al., 2015), so miners must use the mining algorithm in multiple systems to maximize profit.

Funds can be stolen in a variety of methods, including email links, programs, websites, and plugins (Eskandari, et al., 2018), but the most popular method is website theft. Over the past few years, nearly 4000 cryptocurrencies known as altcoins have been newly developed (Bijmans, et al., 2019). One of the most popular altcoins for mining in the browser since 2018 has recently been renamed Menthe (MINTME) using webchain and cryptonite as proof. This algorithm, called hard memory, requires a lot of disk operations. The classic cryptonite technique was first published in 2013 and consists of three basic aspects: scratchpad implementation, memory-hard loop, and end result calculation, as described in Figure (Pastrana & Guillermo Suarez-Tangil, 2019).

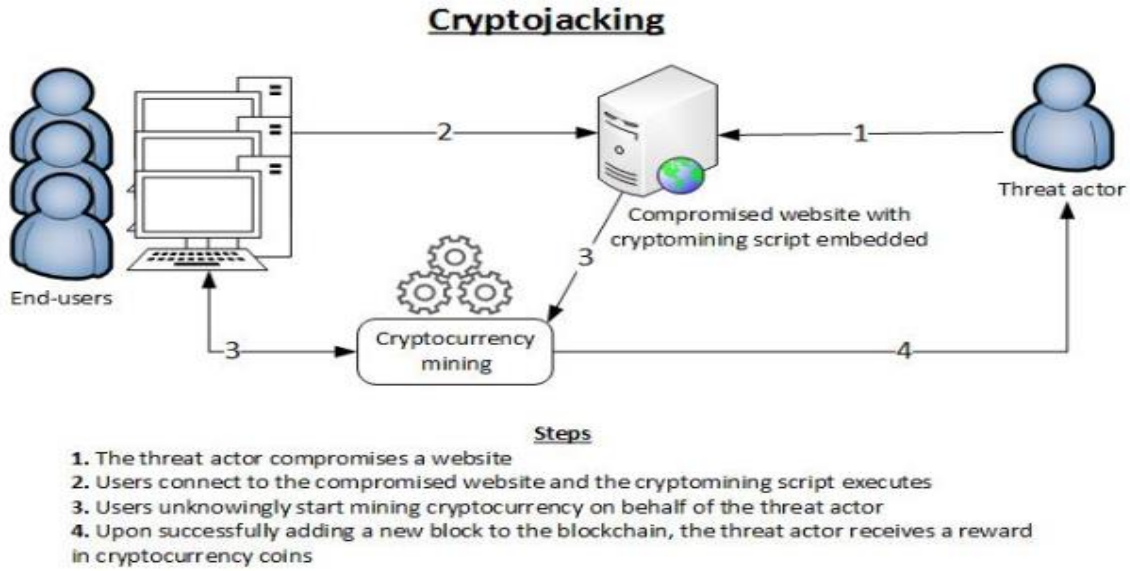


**Figure 1: Cryptojacking process layout**

Conventional CPUs are the main target when attackers use this type of algorithm, as they have about 2MB of desired memory value, which is readily available in the archive. The emergence of the cryptonite algorithm has increased the number of attacks on cryptojacking websites, as the process is performed on common CPUs, which are the most widely available programs worldwide.

The mining process is explained by (Gilson, 2013) where the miners create a community designated as mining pool. At this price they split the revenue for the tasks they did. The workload on the lake is still highly distributed, with miners finding difficult puzzles with a higher load and vice versa. Amongst the most effective things employed to analyze kernel-level activities like CPU efficiency estimator, tracepoint, and kprobe is a software named 'Perf' established by (Gilad, et al., 2017). Researchers should examine core technology aspects with this program.

## 1.2. Research Rationale

The manner people acquire income digitally is evolving around the globe. Previously, it was the sole way to enter Websites that contain ads (ads) embedded in web pages and in some cases make websites annoying and, in some cases, inoperable, culminating to the proliferation of web browsers. Despite advertising has always been on trend, regulators have begun to hunt for alternative strategies to commercialize their services and replenish a few of the squandered wealth.

Cryptocurrencies prove useful in this place. Cryptocurrencies have been around for over 10 years. Bitcoin, was originally established in 2008 (Nakamoto, n.d.), but today there are many others (Tschorsch & Björn Scheuermann, 2016), (Bonneau, et al., 2015). Cyber criminals hijack websites for entertainment or profit. In the latter case, profit is usually made by selling the correct information to companies that can benefit from it. Therefore, the best target is the average consumer. If cybercriminals successfully infect web sites and mine embedded ones without finding them, they might invest millions of clicks a day receiving pages for profit. Mining cryptocurrency every second on web pages is illegal. In fact, it is one of the commercials. Cryptocurrency virus, of course, does not request authorization from victims.

## 1.3. Research Questions

This research report addresses following questions:

1. Does the nature of the dataset utilise for the machine learning model training, testing and evaluation affects the detection of cryptojacking?
2. What is the target attribute considered while analysing the cryptohijacking?
3. Which model will perform the best for the selected dataset and target attribute?
4. Which evaluation metric has been considered for the model evaluation?

## 1.4. Research Objective

A tracker analyzes activities in actual moment whenever crimes happen, bringing a new way for identifying cryptojacking relying on CPU utilization of accessed internet sites. The computer program receives a collection of CPU variables from the scanner. Certain metrics like accuracy and memory are exhibited by integrating a collection of CPU monitoring characteristics and apps with deep learning.

Spending CPU time and generating a warning whenever a given benchmark is surpassed is one approach to identify cryptojacking fraud from internet sites. It yields unfavorable effects due to two factors. For starters, certain online programs, such as video streaming or conference apps, use a lot of CPU power over a considerable length of time. Such applications have a very convenient method of producing false positives. Secondly, the malware code causes the CPU to run at a reduced rate in order to stay underneath the restriction. This contributes to deception. Instead of considering CPU time as a standard figure, these issues are handled by combining characteristics depending on a variety of CPU characteristics.

## 1.5. Structure of the report

The research analysis is organized into two significant parts namely literature review and the research result analysis. Literature review section consists research objective, research overview, research question and the related work. Research result analysis section consists of machine learning training, testing and implementation along with conclusion and comparative analysis of the models used.

# 2. Related Work

Bitcoin greatly reduces the processing speed interruptions necessary to validate and add transactions to the chain. Those who do not have specialized software can assist. Providing small equipment (E.g., smartphones, laptops, desktops) or more powerful mining systems (e.g., offices, servers) may not have adequate value. It deprives clients and allows just a small number of people around the globe to contribute and get subsequent benefits. This trend has shifted after the introduction of other CPU-based cryptocurrencies. Sometimes, when looking at a mine filled with diggers, a gold miner has to keep only the pick-ax on his shoulders and find a promising new edge. Responding to the "gold rush" will reintroduce several of the assaults that were rendered obsolete by bitcoin. Cryptojacking is the word for this form of threat. Tangible assets are "borrowed" by attackers and dishonest personnel who seek to receive money to conduct the mining operation. Various alternatives have recently been presented in response to the concerns, with the goal of developing mitigating strategies.

## 2.1. Cryptojacking Analysis

One of the first methods used to detect cryptocurrency theft was to analyze statistical signatures similar to other types of malwares (Nakamoto, n.d.). Many solutions, such as (FranciscoPereira, et al., 2009) and (Kim, et al., 2018), use static methods, discover mining activities and avoid malicious websites. This approach has proven ineffective compared to cryptojacking, (Li, et al., 2019) due to the use of obscure methods to avoid identification. The detection of crypto theft was done through the first step in the application of machine learning techniques (Gilson, 2013). The authors presented a test study in which Flexible Opcode Analysis successfully launches browser-based crypto-mining innovation. The proposed model would differentiate between cryptomining sites and armed sites (e.g., malicious sites). Where crypto-mining code is injected, crypto-mining sites are reduced (e.g., crypto-mining sites). In (Muñoz, et al., 2019), the authors introduced a way to detect malicious browser behavior.

| Crytocurrency | Type | Client | Version |
|---|---|---|---|
| Bitcoin | Full Node | Bitcoin Core | 0.17.0 |
| Bitcoin | Miner | Bfgminer | 5.5.0 |
| Monero | Full Node | Lithium Luna | 0.12.3.0 |
| Bytecoin | Full Node | Bytecoin Wallet | 3.3.2 |
| Bytecoin / Monero | Miner | XMrig | 2.8.1 |

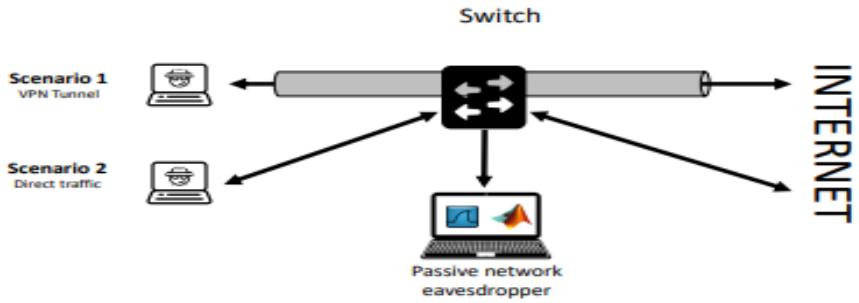


**Figure 2: Analysis of crypto- currencies for cryptohijacking traffic evaluation**

Heap snapshot features and stock elements are extracted and automatically separated using a recurring neural network (RNN). Analyzing 1159 malignant specimens, the test results show that the proposed specimen identifies the true specimens in the mines, with 93% accuracy if not hung up (Kim, et al., 2018) After identifying a set of natural cryptocurrency scripts such as hash-based duplication and standard call stack, a behavior-based finder called CMtracker was introduced (Han, et al., 2018). The researchers make a suggestion Capjack, a machine-based search engine that can detect malicious cryptocurrency mining activity in the browser. This solution uses CapsNet, a machine learning algorithm that simulates the organization of a biological nerve. CapJack utilizes system features such as CPU, memory, disk and network usage using a host-based solution with an 87% authentication rate (Li, et al., 2019).

## 2.2. Network Classification using Machine learning techniques

In recent years there has been a greater focus on network traffic segmentation, which is likely Packet analysis of the classic method of solving certain problems in network management (Gomes, et al., 2021), (Wagner & Paolo Soto, 2002) (for example, creating network profiles

to monitor and manage network traffic in real time), as well as network security (Saad, et al., 2018) (e.g., machine learning applications for detection) will be head and payload check. In addition to the high accuracy of this method, a large amount of data processed, along with the consumer privacy issues raised by this method, prompted the research community to explore various strategies.

In addition, having encrypted traffic does not imply a payload test, which has become commonplace today. Learning algorithms are explored in both real-time IP congestion control as well as downstream evaluation of pre-captured communication in a potential development guideline. The influence of deep learning systems on internet traffic segmentation became one of the earliest challenges (Kharraz, et al., 2019). The researchers apply uncontrolled artificial intelligence techniques to exploit statistical velocity components to dynamically divide network congestion. Examine and assess the impact of individual factor, such as forward-puck-lane-wise, backward-puck-lane-wise, backward-bytes, forward-peklen-mean, forward-bytes, backward-pect-len-mean, and time.

Forward-IAT-Mean, multiple traffic tracks collected from various Internet sites were used to assess the effectiveness of the method adopted. In (Petrov, et al., 2020), the authors examine the major solutions to isolate machine-based IP traffic proposed in the literature. For unencrypted deep packet inspection, numerous asynchronous deep learning methods (such as autoclass, anticipation optimality, decision tree, and naive base) deliver good precision (up to 99 percent) in traffic for multiple internet uses.In (Gilson, 2013) the authors tested various machine learning algorithms to differentiate network traffic flow based on computational accuracy and cost. Specifically, they investigated the use of three monitor algorithms (ie, Boisian networks, decision trees, and multilayer perceptron): peer-to-peer (P2P), web (HTTP), content delivery (akamai), bulk (FTP) service (DN) and mail (SMTP). Their results show decision trees have a higher accuracy and a higher level of discrimination than Boazian networks. However, decision trees require a lot of time to build and the risk of having the wrong or small amount of training data is high.

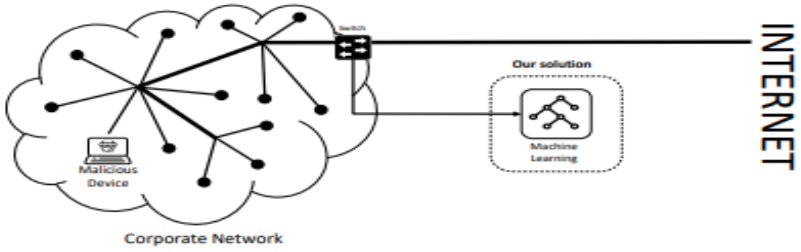


**Figure 3: Network traffic analysis using graphical nodal approach**

As a consequence, authors recommend using a methodical way of creating structured learning groups that provide the highest prediction performance. The first method for protected traffic monitoring derives from (Tschorsch & Björn Scheuermann, 2016). To split SSH activity from non-SSH activity into distinct traffic pathways, the researchers apply a range of classification algorithms (e.g., Adaboost, Vector Support Machine, Nive Bayesian, RIPPER, and C4.5). Their findings suggest that the version generated by the C4.5 algorithm surpasses everyone else when flow-based characteristics are included. Other contributions to this category include (Li, et al., 2019) and (Han, et al., 2018). Using the built-in Android Machine Traffic Network-based monitoring strategy, the authors show that external attackers can detect user-specific actions in their mobile application. Using random forest taxonomies,

they can understand not only the app used by the target user, but also the specific functions it performs (e.g., sending emails, sending messages, home refreshing, etc.)

Cryptojacking, as mentioned earlier, is a good idea and also an alternative to advertising - that can be badly used to steal computing. Similar attacks have increased over the past year. According to the 2018 Symantec Security Annual Report (Hardcastle, 2018) it rose to 85% in 2017 alone, a surprising increase that allows it to compete directly with the most widespread attacks today, namely ransomware. Cryptojacking soared in 2017, 85% according to the 2018 Symantec Security Annual Report (Hardcastle, 2018) then declined and is now rising again.

Kim et al. (Kim et al., 2018) Emphasize that the real trick to spot a bitcoin extraction website is to keep an eye on the implementation of the mineral processing capabilities listed below: WebAssembly; Web workers (a large number); etc.

Kim et al. (2018) developed MiningHunter, a platform for monitoring bitcoin programs. Each visited webpage, information, all executed JavaScript, including actual WebSocket connection are all stored. The information is evaluated and contrasted, utilizing structures and variables looking for frequent characters and procedures with repeated similar fingerprints. Based on the earlier phase, the identities were compared to see if any of them belonged to a certain mining organization.

To summarize, various investigations employed machine learning techniques to predict dangers in computing technologies; various approaches and suggestions for further studies were recommended, however the emphasis was on accuracy rate, precision, recall, and f1-score. Several computer strategies for accuracy, precision, recall, and f1-score will be performed on an openly accessible datasets in this investigation.

# 3. Research Methodology

The strategy utilized in this publication is an outgrowth of earlier research in the topic. The evaluation and investigation of crucial transactions actually occurring for discovery is studied in the current procedures. The goal of this study is to figure out which method has the highest accuracy. Naive Bayes, K-Nearest Neighbor, Decision Tree, and Random Forest are the multiple machine learning techniques used in the report.

## 3.1. Four Machine Learning Models

The Nave Bayes method is a supervised machine learning approach for addressing categorization issues that is predicated on the Bayes theorem. It is a basic and efficient probabilistic classifier that aids in the development of rapid neural network models capable of making accurate recommendations.

The K-Nearest Neighbour strategy is focused on the Supervised Classification algorithm and is among the most basic Machine Learning techniques. The KNN method believes that the novel specific instance and existing situations are equivalent and places the legal dispute in the group which is most compatible with the existing classifications.

The K-NN technique accumulates any observational evidence and qualifies a novel set of data depending on its resemblance to the known information. This implies that new information can be effectively sorted into a well-defined group using the K-NN method.

Random Forest is a classification algorithm that includes a variety of decision trees on different subgroups of a particular dataset and chooses the mean to enhance the predicted performance of that information," according to the description. Rather than focusing on a single decision tree, the random forest collects the forecasts from every tree and anticipates the correct outcome solely on the overwhelming choices of forecasts. The bigger the quantity of trees in the forest, the more accurate it is and the concern of errors is avoided.

Decision Tree is a classification algorithm that may be employed to solve either categorization or regression difficulties, however it is most commonly employed to solve categorization issues. Nodes in the network indicate information attributes, branching provide prediction model, and every leaf node provides the conclusion in this tree-structured algorithm. It's termed a decision tree since, like a tree, it begins with the cluster head and grows into a tree-like architecture with additional sections.

## 3.2. About Dataset:

Cryptojacking is the unlawful mining of cryptocurrencies on somebody else 's device. As unknowing individuals utilize their machines ordinarily, the crypto-mining software runs in the meantime. Delayed efficiency or implementation gaps are the only indications they'll perceive. However, to the increasing computing capacity of data centers and numerous improperly designed host configurations, cybercriminals have recently shifted their objectives from desktop machines to cloud storage.

This recently founded collection is aimed at analyzing a service instance's productivity throughout a crypto-jacking assault and encouraging innovative ways to identify such intrusions using performance measurements.

The time-series performance information throughout a malware assault and during zero malware assaults are included in the anormal and normal records, correspondingly. The whole sample contains a combination of those two datasets.

# 4. Design Specification

Python language is being used to test which range fits the training values so that the selected divider can be used to divide the test values. Dividing values by ranges (1 or 0) is defined only by training values: 1 means that there is an active minor and 0 means that none of the miners are active. The algorithm needs to know the result of what it is testing to consistently sample and separate unknown values. To select the range that best suits the data, the maximum number of dividers in Python that can be used has been applied here.

Their accuracy rating will be considered when choosing the best option. It started by training the algorithm using k-fold cross validation, because after careful testing it is noted that the validation method is often used and gives very disappointing results. This training model is called k-fold cross-validation, where k represents only the parameter and the number of groups into which the data is divided. Different authentication method in this approach is better than others because it helps to make a better and more realistic assessment of how the model behaves compared to the data that can be used in the training set. In short, this process begins by randomly dividing the view set into k groups of approximately equal size. The first

group is considered the confirmation set and the method is similar to the rest of the k-1 wraps.

During the experiment, examined several class dividers were examined to find the one that provided the best results. In most cases the first two algorithms are used, allowing the label to be applied to multiple values (corresponding bags), while the last two are used in the same case.

## 4.1. Techniques

Machine learning is a technological method that offers robots intellectual capability and enables systems act like a human brain. In this process, a piece of statistics is being utilized to educate the system, which is referred to as training the device, and then the device is evaluated using a procedure known as testing the dataset. These technologies can be employed to any world of engineering with a bit reduced complexity. Methodologies of machine learning have been divided into two categories: supervised learning and unsupervised learning. Whenever the procedures to be implemented are previously spelled out and a type of documents is utilized to instruct the device a certain technique, supervised learning is employed, but unsupervised learning is deployed whenever researchers need the device to investigate on its own and identify interesting insights (GeertMeyfroidt, et al., 2019).

## 4.2. Sklearn framework

Sklearn, commonly referred as the Scikit-learn architecture, is a user-friendly structure that includes a number of useful features like categorization, modeling, and segmentation, as well as pre-processing and assessment algorithms. This is a well-known and simple-to-integrate expansive platform. It aids in the processing of large datasets, and can be used to estimate the value of an impending sporting event, for instance. It is concise to understand and apply because a simple network can be developed and educated using only three bits of code.

The metric function of the scikit-learn package can be employed to create the matrix product, that is commonly exploited to measure the effectiveness of a method or strategy. It includes a huge collection of algorithms that may be leveraged to conveniently execute machine learning techniques (Loobuyck, 2020)

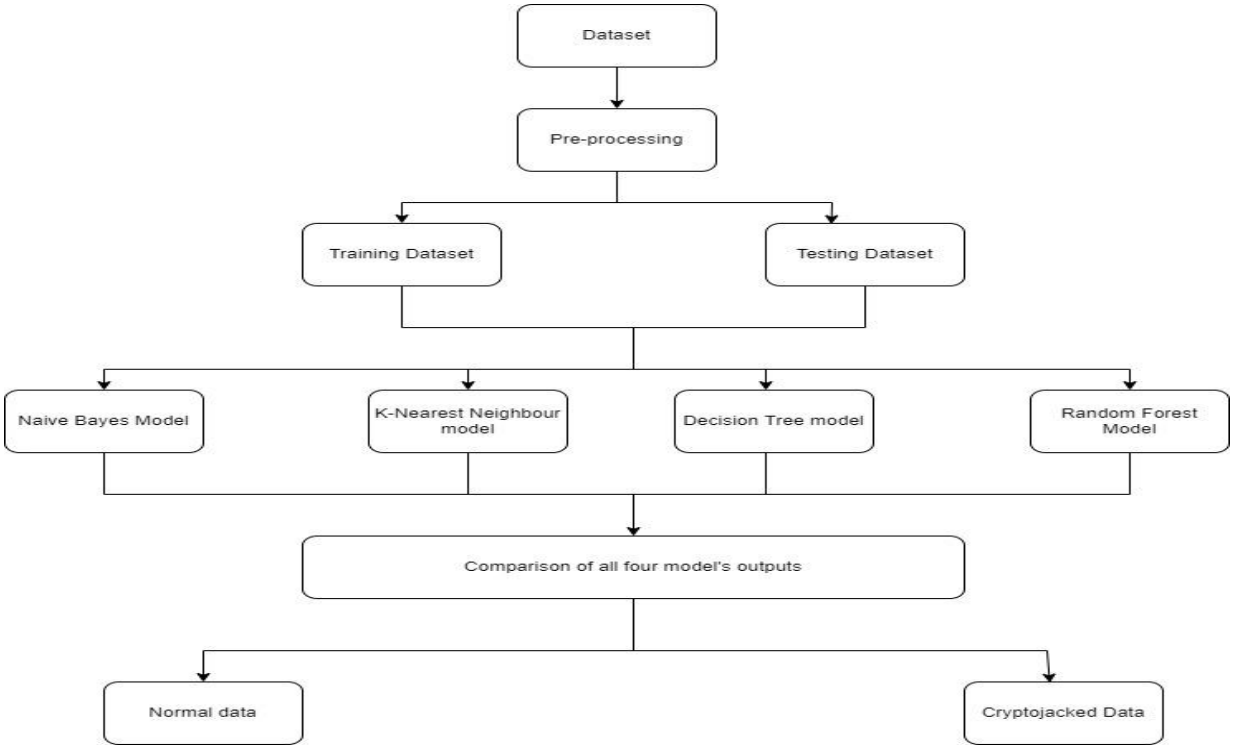


**Figure 4: Framework Diagram**

# 5. Implementation

## 5.1. Setup

Python is used for both execution and pattern generation. The algorithm was developed and executed using Google Colab, that is a Google Research tool. Colab was chosen for this study because it operates on a Google cloud and gives researchers unlimited exposure with increased throughput.

## 5.2. Tools Used

| Tools, Language, Libraries Used | Functionality |
|---|---|
| Python3 (language) | An expansive, interpretive elevated software package for functional programming. It's extensively employed for data mining algorithms because it has a large number of packages and algorithms for doing so (Rolon-M´erette, et al., 2020). |
| pandas (library) | A Python module that may be leveraged to manipulate and analyze information (Millman & Michael Aivazis, 2011). |
| numpy (library) | A widely employed python library for manipulating with collections (Millman & Michael Aivazis, 2011). |

| | |
|---|---|
| sklearn (library) | A machine learning package that is widely adopted to create various strategies (Millman & Michael Aivazis, 2011). |
| matplotlib (library) | Actively planning 2D arrays with a graphical python module (Millman & Michael Aivazis, 2011). |

**Table 1: Tools, Technologies and Libraries used**

## 5.3. Steps to implement the machine learning models:

**1. Data Pre-processing step:**

This stage involves pre-processing or analyzing the information so that it may be utilized effectively in the program. "dataset = pd.read csv('user data.csv')" is used to import the set of data into the application.

The imported data is split into two sections: training and testing.

**2. Fitting machine learning model to the training set:**

The model is applicable to the Training set once it has been pre-processed.

**3. Prediction of the test set result:**

A unique predictor parameter is established for forecasting the test set result, and the predict method is utilized to generate the forecasts.

**4. Creating Confusion matrix:**

The Confusion matrix is used to test the computational performance of the models.

**5. Visualizing the training set result:**

The training set result is depicted by using specific machine learning algorithms (javatpoint, 2022).

Upon effectively executing the research study, the expected outcome is the accuracy rate achieved. The performance of the classification algorithm is represented by this statistic. The acquired precision rate is 99 percent, which is a respectable figure.

# 6. Evaluation

This is a vital step wherein each of the methodologies and approaches performed on the sample are compared, and the methodology chosen is proven and validated for usage in deployment. The productivity and consequences among all algorithms are analysed throughout this pilot study, and the maximum performance of the suggested method is determined utilizing clustering algorithms that determine the prediction performance of assaults employing the information. The confusion matrix displays the effectiveness of the algorithm in deep learning. Numerous criteria like accuracy, precision, recall, and F1 score are explored to evaluate such classifiers.

## 6.1. Model 1: Naïve Bayes model

### 6.1.1. Experiment 1: If the entire dataset has been chosen.

The dataset comprises 95312 rows and 82 columns, and the entire dataset was evaluated over predictor variable, yielding an accuracy of 43% applying the Naive Bayes model.

```
Confusion Matrix :
[[ 0  0  0 ...  0  0  0]
 [ 0  9  2 ...  0  0  0]
 [ 0 32 10 ...  0  0  0]
 ...
 [ 0  0  0 ...  5  1  0]
 [ 0  0  0 ...  3  0  5]
 [ 0  0  0 ...  0  0  2]]
Accuracy Score : 0.43700123994531526
```

**Figure 5: Evaluation metrics for Naive Bayes model for complete data**

```
Report :
              precision    recall  f1-score   support

         1.0       0.00      0.00      0.00         1
        89.0       0.05      0.64      0.09        14
        90.0       0.03      0.04      0.04       275
        91.0       0.92      0.95      0.94      3027
        92.0       1.00      0.38      0.55        24
        93.0       0.00      0.00      0.00        87
        94.0       0.28      0.97      0.43      1311
        95.0       0.00      0.00      0.00       916
        96.0       0.63      0.97      0.76      5848
        97.0       0.87      0.56      0.68      1437
        98.0       0.00      0.00      0.00       514
        99.0       0.00      0.00      0.00       323
       100.0       0.33      0.00      0.00      3482
       101.0       0.67      0.90      0.77      1719
       102.0       0.00      0.00      0.00        72
       103.0       0.52      0.82      0.64       236
       104.0       0.00      0.00      0.00       729
       105.0       0.00      0.00      0.00       350
```

| | | | | |
|---|---|---|---|---|
| 106.0 | 0.36 | 0.42 | 0.39 | 874 |
| 107.0 | 0.00 | 0.00 | 0.00 | 2143 |
| 108.0 | 0.00 | 0.00 | 0.00 | 81 |
| 109.0 | 0.00 | 0.00 | 0.00 | 200 |
| 110.0 | 0.00 | 0.00 | 0.00 | 1285 |
| 111.0 | 0.61 | 0.01 | 0.01 | 3835 |
| 112.0 | 0.07 | 0.34 | 0.11 | 188 |
| 113.0 | 0.50 | 0.02 | 0.04 | 45 |
| 114.0 | 0.00 | 0.00 | 0.00 | 133 |
| 115.0 | 0.08 | 0.76 | 0.14 | 402 |
| 116.0 | 0.22 | 0.86 | 0.35 | 626 |
| 117.0 | 0.00 | 0.00 | 0.00 | 1096 |
| 118.0 | 0.00 | 0.00 | 0.00 | 84 |
| 119.0 | 0.00 | 0.00 | 0.00 | 33 |
| 120.0 | 0.00 | 0.00 | 0.00 | 6 |
| 121.0 | 0.02 | 0.50 | 0.04 | 34 |
| 122.0 | 0.42 | 0.83 | 0.56 | 6 |
| 123.0 | 0.00 | 0.00 | 0.00 | 15 |
| 124.0 | 0.29 | 1.00 | 0.44 | 2 |
| accuracy | | | 0.44 | 31453 |
| macro avg | 0.21 | 0.30 | 0.19 | 31453 |
| weighted avg | 0.43 | 0.44 | 0.35 | 31453 |

**Figure 6: Classification report for Naive Bayes model for complete data**

## 6.1.2. Experiment 2: If only 8 parameters has been chosen.

The dataset comprises 95310 rows and 6 columns, and just 8 parameters are evaluated over predictor varia-le, yielding an accuracy of 52% applying the Nave Bayes model.

```
Confusion Matrix :
[[0 0 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
Accuracy Score : 0.5208406193367883
```

**Figure 7: Evaluation metrics for Naive Bayes model for 8 selected column datasets**

## 6.1.3. Experiment 3: If only 16 parameters has been chosen.

The dataset comprises 95310 rows and 8 columns, and just 16 parameters are evaluated over predictor variable, yielding an accuracy of 82% applying the Nave Bayes model.

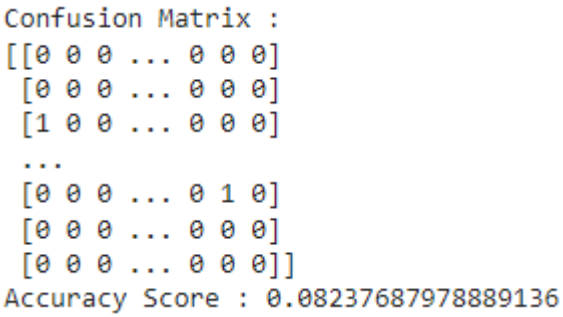


**Figure 8: Evaluation metrics for Naive Bayes model for 16 selected column datasets**

## 6.2. Model 2: K-Nearest Neighbour model

### 6.2.1. Experiment 1: If the entire dataset has been chosen.

The dataset comprises 95312 rows and 82 columns, and the entire dataset was evaluated over predictor variable, yielding an accuracy of 89% applying the K-Nearest Neighbour model.

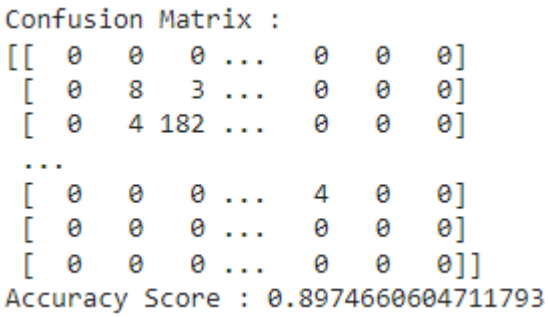


**Figure 9: Evaluation metrics for KNN model for complete data**

Report :

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 1.0 | 0.00 | 0.00 | 0.00 | 1 |
| 89.0 | 0.67 | 0.57 | 0.62 | 14 |
| 90.0 | 0.80 | 0.66 | 0.73 | 275 |
| 91.0 | 0.97 | 0.99 | 0.98 | 3027 |
| 92.0 | 1.00 | 1.00 | 1.00 | 24 |
| 93.0 | 0.67 | 0.61 | 0.64 | 87 |
| 94.0 | 0.97 | 0.97 | 0.97 | 1311 |
| 95.0 | 0.82 | 0.78 | 0.80 | 916 |
| 96.0 | 0.96 | 0.97 | 0.97 | 5848 |
| 97.0 | 0.96 | 0.97 | 0.96 | 1437 |
| 98.0 | 0.94 | 0.96 | 0.95 | 514 |
| 99.0 | 0.78 | 0.64 | 0.71 | 323 |
| 100.0 | 0.96 | 0.97 | 0.96 | 3482 |
| 101.0 | 0.97 | 0.97 | 0.97 | 1719 |
| 102.0 | 0.81 | 0.89 | 0.85 | 72 |
| 103.0 | 0.96 | 0.92 | 0.94 | 236 |
| 104.0 | 0.97 | 0.98 | 0.98 | 729 |
| 105.0 | 0.95 | 0.89 | 0.92 | 350 |
| 106.0 | 0.92 | 0.83 | 0.87 | 874 |
| 107.0 | 0.83 | 0.96 | 0.89 | 2143 |
| 108.0 | 0.88 | 0.62 | 0.72 | 81 |
| 109.0 | 0.52 | 0.21 | 0.30 | 200 |
| 110.0 | 0.74 | 0.57 | 0.64 | 1285 |
| 111.0 | 0.78 | 0.89 | 0.83 | 3835 |
| 112.0 | 0.72 | 0.56 | 0.63 | 188 |
| 113.0 | 0.70 | 0.47 | 0.56 | 45 |
| 114.0 | 0.90 | 0.62 | 0.73 | 133 |
| 115.0 | 0.66 | 0.53 | 0.59 | 402 |
| 116.0 | 0.80 | 0.78 | 0.79 | 626 |
| 117.0 | 0.82 | 0.82 | 0.82 | 1096 |
| 118.0 | 0.83 | 0.60 | 0.69 | 84 |
| 119.0 | 0.20 | 0.03 | 0.05 | 33 |
| 120.0 | 1.00 | 0.83 | 0.91 | 6 |
| 121.0 | 0.50 | 0.09 | 0.15 | 34 |
| 122.0 | 1.00 | 0.67 | 0.80 | 6 |
| 123.0 | 0.00 | 0.00 | 0.00 | 15 |
| 124.0 | 0.00 | 0.00 | 0.00 | 2 |
| accuracy | | | 0.90 | 31453 |
| macro avg | 0.76 | 0.67 | 0.70 | 31453 |
| weighted avg | 0.89 | 0.90 | 0.89 | 31453 |

**Figure 10: Classification report for KNN model for complete column dataset**

### 6.2.2. Experiment 2: If only 8 parameters has been chosen.

The dataset comprises 95310 rows and 6 columns, and just 8 parameters are evaluated over predictor variable, yielding an accuracy of 80% applying the K-Nearest Neighbour model.

```
Confusion Matrix :
[[ 0  0  0 ...  0  0  0]
 [ 0  4  5 ...  0  0  0]
 [ 0  3 17 ...  0  0  0]
 ...
 [ 0  0  0 ...  4  0  0]
 [ 0  0  0 ...  0  0  0]
 [ 0  0  0 ...  0  0  0]]
Accuracy Score : 0.8022128254856452
```

**Figure 11: Evaluation metrics for KNN model for 8 selected column datasets**

### 6.2.3. Experiment 2: If only 16 parameters has been chosen.

The dataset comprises 95310 rows and 8 columns, and just 16 parameters are evaluated over predictor variable, yielding an accuracy of 85% applying the K-Nearest Neighbour model.

```
Confusion Matrix :
[[0 0 0 ... 0 0 0]
 [1 0 0 ... 0 0 0]
 [1 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]]
Accuracy Score : 0.08533367246367596
```

**Figure 12: Evaluation metrics for KNN model for 16 selected column datasets**

## 6.3. Model 2: Decision Tree model

### 6.3.1. Experiment 1: If the entire dataset has been chosen.

The dataset comprises 95312 rows and 82 columns, and the entire dataset was evaluated over predictor variable, yielding an accuracy of 99% applying the Decision Tree model.

```
Confusion Matrix :
[[  0   0   0 ...   0   0   0]
 [  0  14   0 ...   0   0   0]
 [  0   0 275 ...   0   0   0]
 ...
 [  0   0   0 ...   5   0   0]
 [  0   0   0 ...   1  12   1]
 [  0   0   0 ...   0   1   0]]
Accuracy Score : 0.998410326518933
```

**Figure 13: Evaluation metrics for Decision Tree model for complete column dataset**

Report :

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 1.0 | 0.00 | 0.00 | 0.00 | 1 |
| 89.0 | 1.00 | 1.00 | 1.00 | 14 |
| 90.0 | 0.99 | 1.00 | 1.00 | 275 |
| 91.0 | 1.00 | 1.00 | 1.00 | 3027 |
| 92.0 | 1.00 | 0.96 | 0.98 | 24 |
| 93.0 | 1.00 | 1.00 | 1.00 | 87 |
| 94.0 | 1.00 | 1.00 | 1.00 | 1311 |
| 95.0 | 1.00 | 1.00 | 1.00 | 916 |
| 96.0 | 1.00 | 1.00 | 1.00 | 5848 |
| 97.0 | 1.00 | 1.00 | 1.00 | 1437 |
| 98.0 | 1.00 | 1.00 | 1.00 | 514 |
| 99.0 | 1.00 | 1.00 | 1.00 | 323 |
| 100.0 | 1.00 | 1.00 | 1.00 | 3482 |
| 101.0 | 1.00 | 1.00 | 1.00 | 1719 |
| 102.0 | 1.00 | 1.00 | 1.00 | 72 |
| 103.0 | 1.00 | 1.00 | 1.00 | 236 |
| 104.0 | 1.00 | 1.00 | 1.00 | 729 |
| 105.0 | 0.99 | 1.00 | 1.00 | 350 |
| 106.0 | 1.00 | 1.00 | 1.00 | 874 |
| 107.0 | 1.00 | 1.00 | 1.00 | 2143 |
| 108.0 | 1.00 | 1.00 | 1.00 | 81 |
| 109.0 | 0.99 | 0.99 | 0.99 | 200 |
| 110.0 | 1.00 | 1.00 | 1.00 | 1285 |
| 111.0 | 1.00 | 1.00 | 1.00 | 3835 |
| 112.0 | 0.98 | 0.99 | 0.99 | 188 |
| 113.0 | 0.96 | 0.96 | 0.96 | 45 |
| 114.0 | 0.99 | 0.95 | 0.97 | 133 |
| 115.0 | 0.99 | 0.99 | 0.99 | 402 |
| 116.0 | 1.00 | 1.00 | 1.00 | 626 |
| 117.0 | 1.00 | 0.99 | 0.99 | 1096 |
| 118.0 | 0.95 | 1.00 | 0.98 | 84 |
| 119.0 | 1.00 | 0.91 | 0.95 | 33 |
| 120.0 | 0.50 | 0.83 | 0.62 | 6 |
| 121.0 | 1.00 | 0.94 | 0.97 | 34 |
| 122.0 | 0.83 | 0.83 | 0.83 | 6 |
| 123.0 | 0.92 | 0.80 | 0.86 | 15 |
| 124.0 | 0.00 | 0.00 | 0.00 | 2 |
| accuracy | | | 1.00 | 31453 |
| macro avg | 0.92 | 0.92 | 0.92 | 31453 |
| weighted avg | 1.00 | 1.00 | 1.00 | 31453 |

**Figure 14: Classification report for Decision Tree model for complete column dataset**

Hence, this experiment concludes that the selected dataset gives highest accuracy and good PRF score for the Decision Tree model as compared to Naïve Bayes model and KNN model.

### 6.3.2. Experiment 2: If only 8 parameters has been chosen.

The dataset comprises 95310 rows and 6 columns, and just 8 parameters are evaluated over predictor variable, yielding an accuracy of 79% applying the Decision Tree model.

```
Confusion Matrix :
[[ 0  0  0 ...  0  0  0]
 [ 0  5  6 ...  0  0  0]
 [ 0  5 67 ...  0  0  0]
 ...
 [ 0  0  0 ...  4  0  0]
 [ 0  0  0 ...  0  1  0]
 [ 0  0  0 ...  0  0  0]]
Accuracy Score : 0.7921025021460592
```

**Figure 15: Evaluation metrics for Decision Tree model for 8 selected column datasets**

Hence, the above experiment concludes that the selected dataset gives better accuracy for the Decision Tree model as compared to Naïve Bayes model and KNN model.

## 6.4. Model 2: Random Forest model

### 6.4.1. Experiment 1: If the entire dataset has been chosen.

The dataset comprises 95312 rows and 82 columns, and the entire dataset was evaluated over predictor variable, yielding an accuracy of 99% applying the Random Forest model.

```
Confusion Matrix :
[[  0   0   0 ...   0   0   0]
 [  0   9   5 ...   0   0   0]
 [  0   0 275 ...   0   0   0]
 ...
 [  0   0   0 ...   6   0   0]
 [  0   0   0 ...   0  15   0]
 [  0   0   0 ...   0   2   0]]
Accuracy Score : 0.9935141321972467
```

**Figure 16: Evaluation metrics for Random Forest model for complete column dataset**

Report :

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 1.0 | 0.00 | 0.00 | 0.00 | 1 |
| 89.0 | 1.00 | 0.64 | 0.78 | 14 |
| 90.0 | 0.98 | 1.00 | 0.99 | 275 |
| 91.0 | 1.00 | 1.00 | 1.00 | 3027 |
| 92.0 | 1.00 | 0.96 | 0.98 | 24 |
| 93.0 | 0.99 | 1.00 | 0.99 | 87 |
| 94.0 | 1.00 | 0.99 | 1.00 | 1311 |
| 95.0 | 0.99 | 1.00 | 0.99 | 916 |
| 96.0 | 1.00 | 1.00 | 1.00 | 5848 |
| 97.0 | 1.00 | 1.00 | 1.00 | 1437 |
| 98.0 | 1.00 | 0.99 | 1.00 | 514 |
| 99.0 | 0.99 | 0.98 | 0.99 | 323 |
| 100.0 | 1.00 | 1.00 | 1.00 | 3482 |
| 101.0 | 1.00 | 1.00 | 1.00 | 1719 |
| 102.0 | 0.90 | 1.00 | 0.95 | 72 |
| 103.0 | 1.00 | 0.99 | 0.99 | 236 |
| 104.0 | 1.00 | 0.99 | 0.99 | 729 |
| 105.0 | 0.98 | 0.98 | 0.98 | 350 |
| 106.0 | 0.99 | 0.98 | 0.99 | 874 |
| 107.0 | 0.99 | 1.00 | 0.99 | 2143 |
| 108.0 | 0.94 | 0.91 | 0.92 | 81 |
| 109.0 | 0.99 | 0.95 | 0.97 | 200 |
| 110.0 | 0.99 | 0.99 | 0.99 | 1285 |
| 111.0 | 0.99 | 1.00 | 0.99 | 3835 |
| 112.0 | 0.96 | 0.91 | 0.93 | 188 |
| 113.0 | 0.84 | 0.71 | 0.77 | 45 |
| 114.0 | 0.96 | 0.92 | 0.94 | 133 |
| 115.0 | 0.98 | 0.98 | 0.98 | 402 |
| 116.0 | 0.98 | 0.99 | 0.99 | 626 |
| 117.0 | 0.99 | 0.99 | 0.99 | 1096 |
| 118.0 | 0.89 | 0.90 | 0.90 | 84 |
| 119.0 | 0.96 | 0.67 | 0.79 | 33 |
| 120.0 | 0.83 | 0.83 | 0.83 | 6 |
| 121.0 | 0.94 | 0.97 | 0.96 | 34 |
| 122.0 | 1.00 | 1.00 | 1.00 | 6 |
| 123.0 | 0.94 | 1.00 | 0.97 | 15 |
| 124.0 | 1.00 | 0.50 | 0.67 | 2 |
| accuracy | | | 0.99 | 31453 |
| macro avg | 0.95 | 0.91 | 0.92 | 31453 |
| weighted avg | 0.99 | 0.99 | 0.99 | 31453 |

**Figure 17: Classification report for Random Forest model for complete column dataset**

Hence, this experiment concludes that the selected dataset gives highest accuracy as well as highest PRF score for the Random Forest model in comparison to Naïve Bayes model and KNN model.

## 6.4.2. Experiment 2: If only 8 parameters has been chosen.

The dataset comprises 95310 rows and 6 columns, and just 8 parameters are evaluated over predictor variable, yielding an accuracy of 81% applying the Decision Tree model.

```
Confusion Matrix :
[[ 0  0  0 ...  0  0  0]
 [ 0  1 10 ...  0  0  0]
 [ 0  6 58 ...  0  0  0]
 ...
 [ 0  0  0 ...  4  0  0]
 [ 0  0  0 ...  0  0  0]
 [ 0  0  0 ...  0  0  0]]
Accuracy Score : 0.8189043970368486
```

**Figure 18: Evaluation metrics for Random Forest model for 8 selected column datasets**

Due to RAM memory restricting only two machine learning models have been utilized for selected 16 columns dataset. For future work we can buy extra RAM and memory storage for the machine learning model training, testing and evaluation. Hence, this experiment concludes that the selected dataset gives highest accuracy for the Random Forest model in comparison to Naïve Bayes model and KNN model.

## 6.5. Discussion

Through a critical analysis of the results and a comparison within each paradigm in the table beneath, it was determined that the **Random Forest Algorithm** has the best accuracy and PRF rate, trailed by the Decision Tree model, who offers the second-highest accuracy and PRF rate. The accuracy and PRF rate of K-Nearest Neighbour and Naïve Bayes are barely mediocre. Random Forest reveals that cryptojacking can be identified by **99 percent** utilizing CPU usage as a goal characteristic to solve the objectives of this research.

Modifying the dataset, on the other hand, might have led to differences for this study, with varying accuracy and PRF scores. Moreover, alternative methods might have yielded mixed results.

| Algorithm | Accuracy | Precision | Recall | F1-score |
|---|---|---|---|---|
| Naïve Bayes | 43 | 43 | 44 | 35 |
| K-Nearest Neighbour | 89 | 89 | 90 | 89 |
| Decision Tree | 99 | 1 | 1 | 1 |
| Random Forest | 99 | 99 | 99 | 99 |

**Table 2: Comparison of outputs**

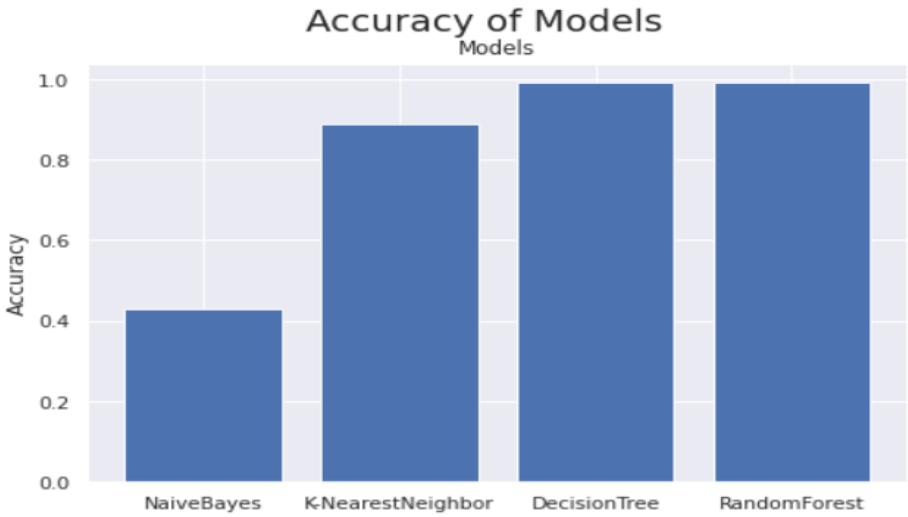


**Figure 19: Accuracy of all the models**

# 7. Conclusion and Future Work

This study demonstrates that by integrating a variety of CPU measures such as Accuracy, it is able to discern significant browser-based cryptojacking. These conclusions, nevertheless, were gathered from assessors who were tested in a controlled setting, which might have influenced outcomes while reviewed in a practical situation.
Various additional discoveries:

- On CPUs with several threads, performing cryptojacking reveals a distinct trend.
- Specified patterns can be detected with amazing precision using a collection of different classifiers.
- On devices with only single sensor, we can determine that the average CPU readings are consistent and display uniform readings throughout all CPU cores whenever the website is executing rapidly.
- Even with various different CPU-intensive applications, recognition on the system performs inadequately.

There could be a broader variety of options in this study than browser-based mining. The study is grounded on CPU parameters and assumptions which are influenced in the same way by minors that do not execute in the web page. Nevertheless, browser-based cryptojacking virus will be the focus of our formative assessment in the long term. Despite showing promising results, the research should not be used as a separate method to verify the presence of cryptogenic miners. It is used only as a cross-validation metric in a virtualized environment and should not be distributed individually to each user for use on their personal computer.

# References


[1] Androulaki, E. et al., 2018. *Hyperledger fabric: a distributed operating system for permissioned blockchains.* s.l., Proceedings of the Thirteenth EuroSys Conference.

[2] Anon., 2019. Machine learning techniques to examine large patient databases. *Best Practice & Research Clinical Anaesthesiology,* 23(1), pp. 127-143.

[3] Apostolaki, M., 2017. *Hijacking Bitcoin: Routing Attacks on Cryptocurrencies.* San Jose, CA, USA, 2017 IEEE Symposium on Security and Privacy (SP).

[4] Bijmans, H. L., Tim M. Booij & Christian Doerr, 2019. *Inadvertently Making Cyber Criminals Rich: A Comprehensive Study of Cryptojacking Campaigns at Internet Scale.* Santa Clara, CA, USA, 28th USENIX Security Symposium..

[5] Bonneau, J. et al., 2015. *SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies.* San Jose, CA, USA, 2015 IEEE Symposium on Security and Privacy.

[6] Carlin, D., Philip O'Kane, Sakir Sezer & Jonah Burgess, 2018. *Detecting Cryptomining Using Dynamic Analysis.* Belfast, Ireland, 2018,16th Annual Conference on Privacy, Security and Trust (PST).

[7] Eskandari, S., Andreas Leoutsarakos, Troy Mursch & Jeremy Clark, 2018. *A First Look at Browser-Based Cryptojacking.* London, UK, 2018 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW).

[8] FranciscoPereira, MatthewBotvinicka & Tom Mitchell, 2009. Machine learning classifiers and fMRI: A tutorial overview. *NeuroImage,* 45(1), pp. S199-S209.

[9] GeertMeyfroidt, Fabian Güiza, Jan Ramon & Maurice Bruynooghe, 2019. Machine learning techniques to examine large patient databases. *Best Practice & Research Clinical Anaesthesiology,* 23(1), pp. 127-143.

[10] Gilad, Y. et al., 2017. *Algorand: Scaling Byzantine Agreements for Cryptocurrencies.* s.l., Proceedings of the 26th Symposium on Operating Systems Principles.

[11] Gilson, D., 2013. *New currency Primecoin searches for prime numbers as proof of work.* [Online]
Available at: https://www.coindesk.com/markets/2013/07/10/new-currency-primecoin-searches-for-prime-numbers-as-proof-of-work/#:~:text=A%20new%20digital%20currency%20has,besides%20its%20subjective%20market%20value.
[Accessed March 2022].

[12] Gomes, G., Luis Dias & Miguel Correia, 2021. *CryingJackpot: Network Flows and Performance Counters against Cryptojacking.* Cambridge, MA, USA, 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA).

[13] Han, R., Vincent Gramoli & Xiwei Xu, 2018. *Evaluating Blockchains for IoT.* s.l., 2018 9th IFIP International Conference on New Technologies, Mobility and Security.

[14] Hardcastle, J. L., 2018. *Symantec Security Report: Cryptojacking Attacks Increased 8,500% in 2017.* [Online]
Available at: https://www.sdxcentral.com/articles/news/symantec-security-report-cryptojacking-attacks-increased-8500-in-2017/2018/03/
[Accessed March 2022].

[15] javatpoint, 2022. *Machine Learning Tutorial.* [Online]
Available at: https://www.javatpoint.com/machine-learning
[Accessed March 2022].

[16] Kharraz, A. et al., 2019. *Outguard: Detecting In-Browser Covert Cryptocurrency Mining in the Wild.* s.l., WWW '19: The World Wide Web Conference.

[17] Kim, H. et al., 2018. *The Other Side of the Coin: A Framework for Detecting and Analyzing Web-based Cryptocurrency Mining Campaigns.* s.l., the 13th International Conference.

[18] Li, K.-C., Xiaofeng Chen, Hai Jiang & Elisa Bertino, 2019. *Essentials of Blockchain Technology.* illustrated ed. s.l.:CRC Press, 2019.

[19] Loobuyck, U., 2020. *Scikit-learn, TensorFlow, PyTorch, Keras… but where to begin?.* [Online]

Available at: https://towardsdatascience.com/scikit-learn-tensorflow-pytorch-keras-but-where-to-begin-9b499e2547d0
[Accessed March 2022].
[20] Millman, K. J. & Michael Aivazis, 2011. *Python for Scientists and Engineers.* s.l., Computing in Science & Engineering.
[21] Muñoz, J. Z. i., José Suárez-Varela & Pere Barlet-Ros, 2019. *Detecting cryptocurrency miners with NetFlow/IPFIX network measurements.* Catania, Italy, 2019 IEEE International Symposium on Measurements & Networking (M&N).
[22] Nakamoto, S., n.d. *Bitcoin: A Peer-to-Peer Electronic Cash System.* [Online] Available at: file:///C:/Users/SNEHAL/Downloads/21260-bitcoin-a-peer-to-peer-electronic-cash-system.pdf
[23] Pastrana, S. & Guillermo Suarez-Tangil, 2019. *A First Look at the Crypto-Mining Malware Ecosystem: A Decade of Unrestricted Wealth.* s.l., Proceedings of the Internet Measurement Conference.
[24] Petrov, I., Luca Invernizzi & Elie Bursztein, 2020. CoinPolice:Detecting Hidden Cryptojacking Attacks with Neural Networks. Volume 2.
[25] Rolon-M´erette, D., Matt Ross, Tadd´e Rolon-M´erette & Kinsey Church, 2020. Introduction to Anaconda and Python: Installation and setup. *The Quantitative Methods for Psychology,* 16(5).
[26] Saad, M., Aminollah Khormali & David Mohaisen, 2018. *End-to-End Analysis of In-Browser Cryptojacking.* [Online]
Available at: https://www.researchgate.net/publication/327549957_End-to-End_Analysis_of_In-Browser_Cryptojacking
[Accessed 2022].
[27] Sirer, E. G., 2013. *Majority Is Not Enough: Bitcoin Mining Is Vulnerable.* s.l., International Conference on Financial Cryptography and Data Security.
[28] Tschorsch, F. & Björn Scheuermann, 2016. Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies. *IEEE Communications Surveys & Tutorials* , 18(3), pp. 2084 - 2123.
[29] Wagner, D. A. & Paolo Soto, 2002. *Mimicry attacks on host-based intrusion detection systems.* s.l., Proceedings of the 9th ACM conference on Computer and communications security.
[30] Wang, W. et al., 2018. *SEISMIC: SEcure In-lined Script Monitors for Interrupting Cryptojacks.* s.l., 23rd European Symposium on Research in Computer Security (ESORICS).