Securing Heterogeneous Smart Home Networks with IoT Cryptojacking Detection

Manal Jain (213050008)
Hrishi Saloi (213050057)
Akash Kumar (213050020)
Siddhant Singh (213050031)

Abstract:

IoT device adoption is resulting in a surge in security risks, including the risky activity of cryptojacking. This assault involves the unauthorized use of computing power to harm devices and networks by mining bitcoins. These types of attacks are particularly susceptible to hitting smart home networks, which frequently have several IoT devices.
[1] suggests a practical method for identifying cryptojacking in IoT devices. This technique uses network traffic analysis to find host-based and in-browser crypto jacking. In this work, we provide an implementation of the techniques proposed in [1] along with an analysis of the results. We also look into the difficulty of implementing cryptojacking detection in new device categories, such as the Internet of Things, and design a number of experiment scenarios to test our detection system against various attacker tactics and network configurations.

The original codebase and dataset of the implementation are available at the following GitHub repo:
https://github.com/manaljain6667/Securing-Heterogeneous-Smart-Home-Networks-with-IoT-Cryptojacking-Detection/tree/main

Introduction:

Internet of Things (IoT) gadgets are being used in smart homes more and more, but this reliance has a price. The hazards related to the security of IoT devices increase along with their quantity. Cryptojacking is one attack type that has grown in popularity in recent years. In this scenario, hackers destroy IoT devices and networks by secretly mining cryptocurrency on them. These assaults have the potential to seriously harm IoT devices, resulting in decreased performance, higher energy consumption, and reduced network bandwidth.

Given the serious consequences of cryptojacking assaults, a reliable and portable detection system is increasingly required to protect smart home networks. Unfortunately, because of the abundance of IoT devices with variable levels of security and computational capacity, smart home networks are especially susceptible to these assaults. This project report tries to address this issue by suggesting a trustworthy and effective mechanism for identifying cryptojacking assaults in smart home networks. Our suggested solution will take into account the distinctive features of IoT devices and smart home networks, and it will be developed to minimise any negative effects on device performance and network capacity.

The suggested method for recognising cryptojacking attacks in smart home networks entails figuring out the distinctive network traffic patterns these assaults produce and using a simple deep learning model to detect them. In this project report, we take into account smart home network configurations where router is connected to devices to gain access to the Internet and where each device is uniquely identifiable by its MAC address. By watching their network activity for a predetermined amount of time, it is possible to identify any hardware that is engaged in cryptocurrency mining.

We use machine learning algorithms that have been previously trained using malicious and benign data to do this. Devices continuously produce network traffic, which needs to be formatted properly. To do this, each packet is first

filtered according to the MAC address of either its source or destination. Once we have a list of packet lengths for each device, we extract metadata from each packet. We compute characteristics that can be used to test and train the method using bursts of 10 packets. Finally, we use this model in multiple tests with various scenarios to assess its usefulness. We seek to minimize any negative effects on device performance and network bandwidth by taking into account the specific characteristics of smart home networks and IoT devices.

Background:

To comprehend the paper's [1] domain, familiarity with cybersecurity, IoT devices, smart homes, and cryptojacking is crucial. Understanding the difficulties of detecting and preventing cryptojacking in diverse smart home networks, as well as the use of lightweight detection methods, is also beneficial.

- **Cryptocurrency Mining**
  Cryptocurrency mining is the process by which new cryptocurrencies are introduced into the market, essential for maintaining the global blockchain ledger. It relies on Proof of Work (PoW) consensus models and cryptographic hash algorithms. Mining is a time-consuming and costly activity, with payouts based on luck. The randomness of hash algorithms prevents miners from predicting values systematically, but it remains a lucrative revenue source for many investors, with higher hardware investment increasing difficulty, cost, and risk.

- **Cryptojacking**
  Cryptojacking is a form of cyberattack where malware is employed to covertly harness the computing power of a victim's device, like a computer, smartphone, or IoT device, for unauthorized cryptocurrency mining. Attackers take control of the device's processing capabilities, leading to increased energy consumption, diminished performance, and potential device damage. Exploiting web app vulnerabilities, malicious downloads/links, and infecting IoT devices exploit their large numbers, limited security features, and update limitations, making them susceptible to cryptojacking. It has 2 types :

    - In-browser cryptojacking: It involves secretly utilizing a victim's web browser to mine cryptocurrency. Attackers insert JavaScript code into websites or online ads that the victim accesses. When executed, the code harnesses the browser's processing power for mining, with the mined cryptocurrency sent to the attacker. The availability of

user-friendly cryptojacking scripts facilitated the rise of in-browser mining, allowing attackers to inject code easily and initiate the mining process.

- Host-based Cryptojacking: It is a cyberattack where malware is installed on a victim's computer to mine cryptocurrency without their consent. The malware is delivered through phishing emails, malicious websites, or software vulnerabilities. Once installed, it operates in the background, utilizing the computer's processing power for mining.

- **Machine Learning Tools:**
  The machine learning algorithms and methods used during the experiments:

- Feature Extraction and Selection Tools: Feature extraction reduces the size of a raw dataset by combining features using property-based functions. This paper utilizes the tsfresh automatic feature calculation tool. Feature selection follows, where less significant features are eliminated based on their P-values, improving the classification process.
- Machine Learning Classifiers: Classification is a way of organizing data into groups to predict the category of new data. We used different models like Logistic Regression, K-Nearest Neighbors (KNN), and Support Vector Machines (SVM) to train our models and get accurate results in our study. (“turkmia”)

Problem Statement:

A typical smart home network is set up, in which many gadgets are linked to the internet by means of a single router. Network packet data transmitted between IoT devices and mining servers is analysed as part of experiments that replicate unauthorised cryptojacking activity. The goal is to find hacked smart home devices that are being used for illegal cryptomining.

Key Challenges:

The accuracy of the detection model relies heavily on a subset of features extracted from network packets, such as timestamp and packet length. However, adversaries can potentially manipulate these crucial features, leading to a decline in the model's accuracy. Adversarial perturbation attacks can be employed to generate manipulated network packets without impacting the functionality of crypto mining protocols at the application level. Perturbation techniques like dummy packets, padding, and splitting are utilized.
Key challenges include the potential manipulation of important features by adversaries, impacting the accuracy of the detection model, and the need to develop robust defenses against adversarial perturbation attacks in the context of cryptojacking detection.

Proposed Solution:

To evaluate the resilience of the model, we conduct perturbation attacks by manipulating network packets through methods like introducing dummy packets, adding padding, splitting packets, and employing obfuscation techniques. The aim is to assess whether these attacks can noticeably reduce the model's F1 score, which was initially reported as 97% in the referenced paper. If successful, we will propose enhancements to the model to mitigate the impact of such attacks and improve its robustness against adversarial manipulations.

Evaluation:

Experiment Setup:

To validate the findings of the referenced work, we replicated the approach by collecting an open-source dataset and training a model specifically designed to detect cryptojacking attacks. We utilized four Machine Learning classifiers (Logistic Regression, KNN, SVM, and GNB) to assess the model's accuracy. Multiple experiments were conducted in different scenarios, following the experimental setups described in the paper. By comparing the results obtained from each classifier, we aimed to identify the best-performing one. The main challenge we focused on, as highlighted in the paper, was the potential vulnerability of the model to adversarial attacks. These attacks could significantly reduce the model's accuracy. We addressed this challenge by creating datasets for various attack scenarios. These included the inclusion of dummy packets, padding packets with extra data, splitting data packets to modify packet lengths and checksum fields, and employing obfuscation proxies to manipulate packet sizes and timings. These attack datasets were combined with the original dataset and fed into our model to assess the model's resilience against these specific attacks. Overall, our evaluation involved replicating the original study, training and testing models with different classifiers, and investigating the model's susceptibility to adversarial attacks by introducing modified datasets for various attack scenarios.

Result: In the limited experimental scenarios that we reproduced from the paper, we achieved a remarkably high accuracy rate of approximately 99% across all the scenarios.

1. **Scenario 1:** In this scenario, a diverse range of devices is deployed in the environment. The objective is to analyze if there are any variations in the detection accuracy when it comes to detecting cryptojacking malware on each specific device.

- Server: Upon providing the relevant data to various machine learning models with different classification algorithms, the following results were obtained:

LogReg

	precision	recall	f1-score	support
malignant	0.99	0.98	0.98	30381
benign	0.98	0.99	0.98	30368
accuracy			0.98	60749
macro avg	0.98	0.98	0.98	60749
weighted avg	0.98	0.98	0.98	60749

KNN

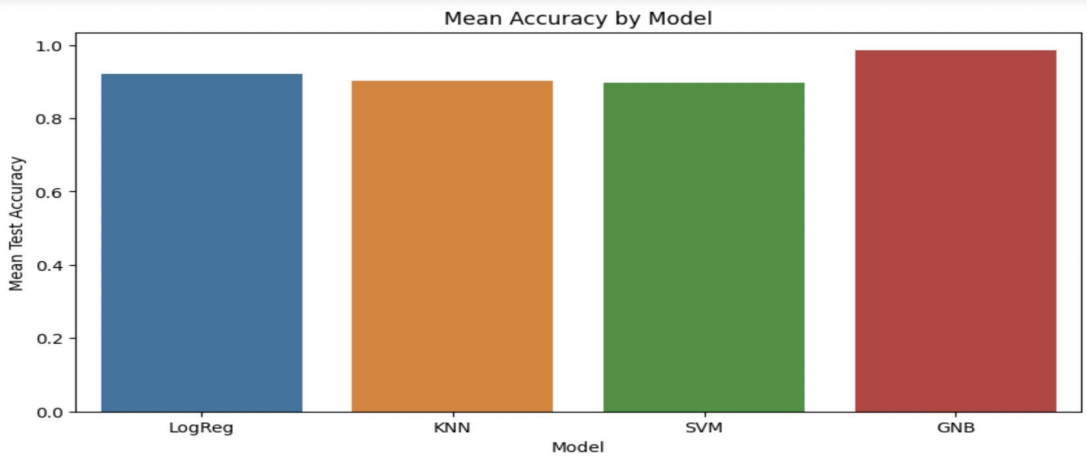
	precision	recall	f1-score	support
malignant	0.99	0.99	0.99	30381
benign	0.99	0.99	0.99	30368
accuracy			0.99	60749
macro avg	0.99	0.99	0.99	60749
weighted avg	0.99	0.99	0.99	60749

SVM

	precision	recall	f1-score	support
malignant	0.99	0.98	0.99	30381
benign	0.98	0.99	0.99	30368
accuracy			0.99	60749
macro avg	0.99	0.99	0.99	60749
weighted avg	0.99	0.99	0.99	60749

GNB

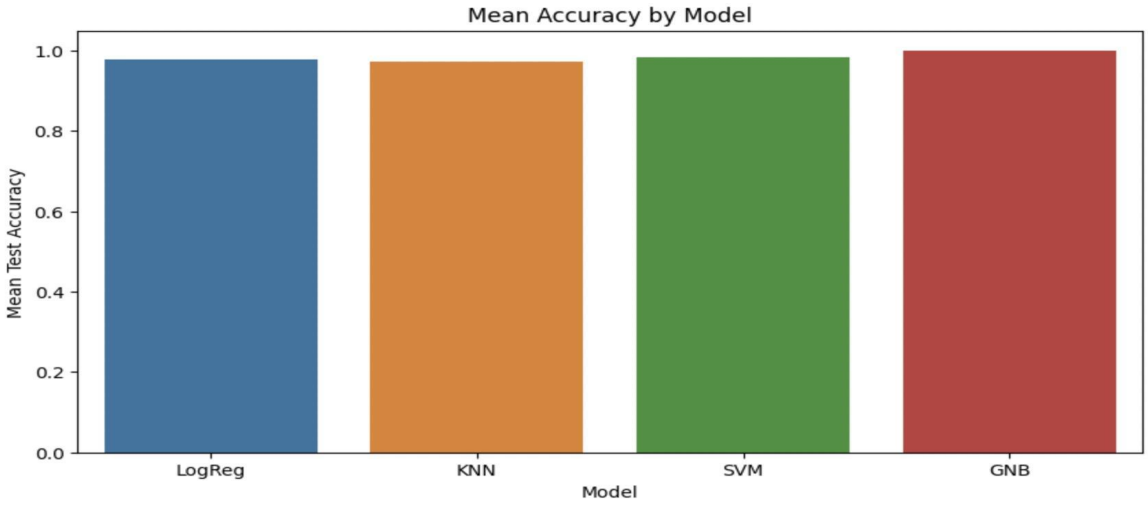
	precision	recall	f1-score	support
malignant	1.00	0.99	0.99	30381
benign	0.99	1.00	0.99	30368
accuracy			0.99	60749
macro avg	0.99	0.99	0.99	60749
weighted avg	0.99	0.99	0.99	60749



- **Laptop**

  After feeding the corresponding data to ML models with different classification algorithms, results obtained are as follows:

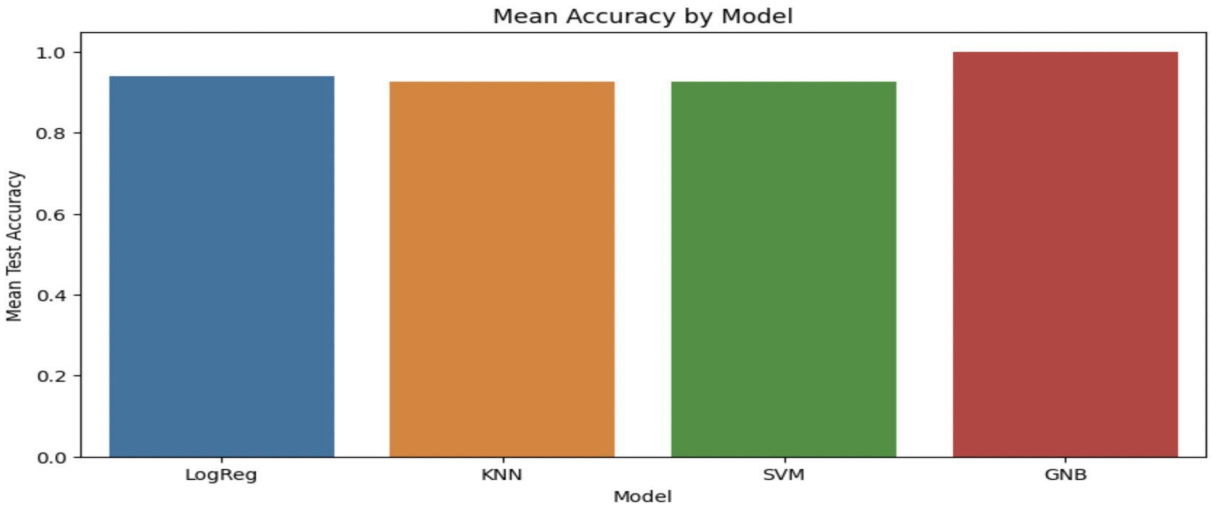
```
LogReg
              precision    recall  f1-score   support

   malignant       0.98      0.98      0.98      5800
      benign       0.98      0.98      0.98      5919

    accuracy                           0.98     11719
   macro avg       0.98      0.98      0.98     11719
weighted avg       0.98      0.98      0.98     11719

KNN
              precision    recall  f1-score   support

   malignant       0.97      0.97      0.97      5800
      benign       0.97      0.97      0.97      5919

    accuracy                           0.97     11719
   macro avg       0.97      0.97      0.97     11719
weighted avg       0.97      0.97      0.97     11719

SVM
              precision    recall  f1-score   support

   malignant       0.98      0.98      0.98      5800
      benign       0.98      0.98      0.98      5919

    accuracy                           0.98     11719
   macro avg       0.98      0.98      0.98     11719
weighted avg       0.98      0.98      0.98     11719

GNB
              precision    recall  f1-score   support

   malignant       1.00      1.00      1.00      5800
      benign       1.00      1.00      1.00      5919

    accuracy                           1.00     11719
   macro avg       1.00      1.00      1.00     11719
weighted avg       1.00      1.00      1.00     11719
```



- **IoT**

  We analyzed the data using different classification algorithms in our machine learning models, and the outcomes are as follows:

```
LogReg
              precision    recall  f1-score   support

   malignant       0.93      0.96      0.94      1539
      benign       0.96      0.93      0.94      1540

    accuracy                           0.94      3079
   macro avg       0.94      0.94      0.94      3079
weighted avg       0.94      0.94      0.94      3079

KNN
              precision    recall  f1-score   support

   malignant       0.93      0.94      0.94      1539
      benign       0.94      0.93      0.94      1540

    accuracy                           0.94      3079
   macro avg       0.94      0.94      0.94      3079
weighted avg       0.94      0.94      0.94      3079

SVM
              precision    recall  f1-score   support

   malignant       0.91      0.95      0.93      1539
      benign       0.95      0.91      0.93      1540

    accuracy                           0.93      3079
   macro avg       0.93      0.93      0.93      3079
weighted avg       0.93      0.93      0.93      3079

GNB
              precision    recall  f1-score   support

   malignant       1.00      1.00      1.00      1539
      benign       1.00      1.00      1.00      1540

    accuracy                           1.00      3079
   macro avg       1.00      1.00      1.00      3079
weighted avg       1.00      1.00      1.00      3079
```



2. Scenario 2 - Profit Strategies:

The attackers commonly use throttle adjustment as an obfuscation method to control the hardware usage on victims' devices. This experiment aims to investigate whether altering the throttle value affects the detection accuracy of our system, as it is an important factor in detecting cryptojacking attacks.

Stealthy attacker (10% throttle)

LogReg

	precision	recall	f1-score	support
malignant	0.95	0.95	0.95	246
benign	0.95	0.95	0.95	248
accuracy			0.95	494
macro avg	0.95	0.95	0.95	494
weighted avg	0.95	0.95	0.95	494

KNN

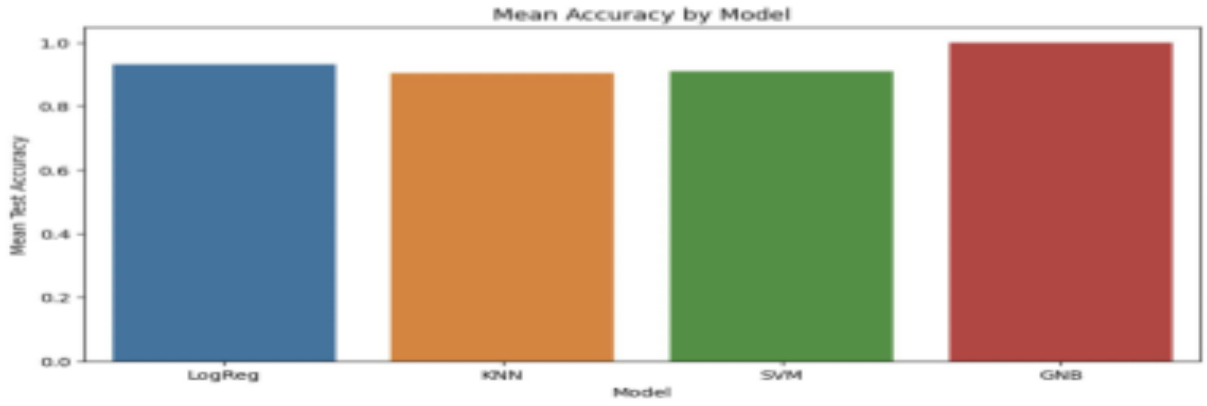
	precision	recall	f1-score	support
malignant	0.89	0.93	0.91	246
benign	0.92	0.88	0.90	248
accuracy			0.90	494
macro avg	0.91	0.90	0.90	494
weighted avg	0.91	0.90	0.90	494

SVM

	precision	recall	f1-score	support
malignant	0.92	0.91	0.91	246
benign	0.91	0.92	0.92	248
accuracy			0.91	494
macro avg	0.92	0.91	0.91	494
weighted avg	0.92	0.91	0.91	494

GNB

	precision	recall	f1-score	support
malignant	1.00	0.99	1.00	246
benign	0.99	1.00	1.00	248
accuracy			1.00	494
macro avg	1.00	1.00	1.00	494
weighted avg	1.00	1.00	1.00	494



- Robust attacker (50% throttle)

LogReg

	precision	recall	f1-score	support
malignant	0.84	0.87	0.85	264
benign	0.85	0.82	0.83	246
accuracy			0.84	510
macro avg	0.84	0.84	0.84	510
weighted avg	0.84	0.84	0.84	510

KNN

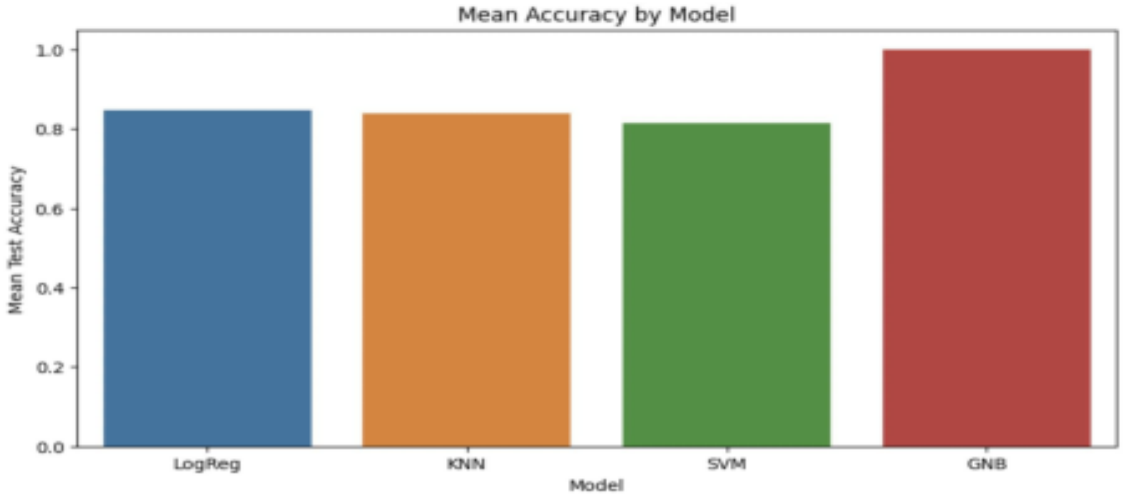
	precision	recall	f1-score	support
malignant	0.85	0.86	0.85	264
benign	0.84	0.84	0.84	246
accuracy			0.85	510
macro avg	0.85	0.85	0.85	510
weighted avg	0.85	0.85	0.85	510

SVM

	precision	recall	f1-score	support
malignant	0.80	0.90	0.85	264
benign	0.88	0.76	0.81	246
accuracy			0.83	510
macro avg	0.84	0.83	0.83	510
weighted avg	0.84	0.83	0.83	510

GNB

	precision	recall	f1-score	support
malignant	0.99	1.00	0.99	264
benign	1.00	0.99	0.99	246
accuracy			0.99	510
macro avg	0.99	0.99	0.99	510
weighted avg	0.99	0.99	0.99	510



- Aggressive attacker (100% throttle)

LogReg

	precision	recall	f1-score	support
malignant	0.98	0.98	0.98	37923
benign	0.98	0.98	0.98	38074
accuracy			0.98	75997
macro avg	0.98	0.98	0.98	75997
weighted avg	0.98	0.98	0.98	75997

KNN

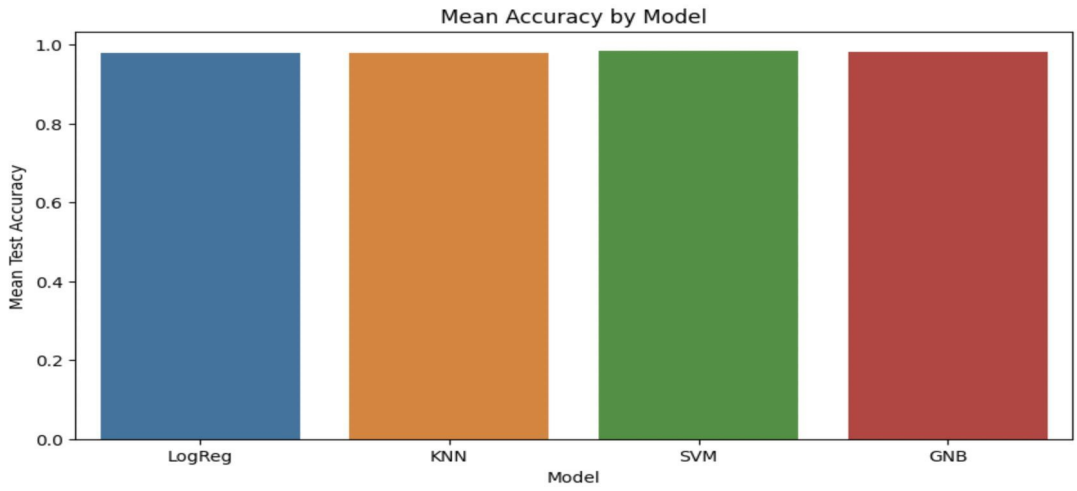
	precision	recall	f1-score	support
malignant	0.98	0.98	0.98	37923
benign	0.98	0.98	0.98	38074
accuracy			0.98	75997
macro avg	0.98	0.98	0.98	75997
weighted avg	0.98	0.98	0.98	75997

SVM

	precision	recall	f1-score	support
malignant	0.99	0.98	0.98	37923
benign	0.98	0.99	0.98	38074
accuracy			0.98	75997
macro avg	0.98	0.98	0.98	75997
weighted avg	0.98	0.98	0.98	75997

GNB

	precision	recall	f1-score	support
malignant	1.00	0.96	0.98	37923
benign	0.97	1.00	0.98	38074
accuracy			0.98	75997
macro avg	0.98	0.98	0.98	75997
weighted avg	0.98	0.98	0.98	75997



3. Scenario 3:

In this experiment, we want to test how well our detection system can tell the difference between in-browser and host-based cryptojacking malware. (“[2103.03851] SoK: Cryptojacking Malware”)

- **In-browser cryptojacking**

LogReg

	precision	recall	f1-score	support
malignant	0.94	0.95	0.94	7673
benign	0.95	0.94	0.94	7624
accuracy			0.94	15297
macro avg	0.94	0.94	0.94	15297
weighted avg	0.94	0.94	0.94	15297

KNN

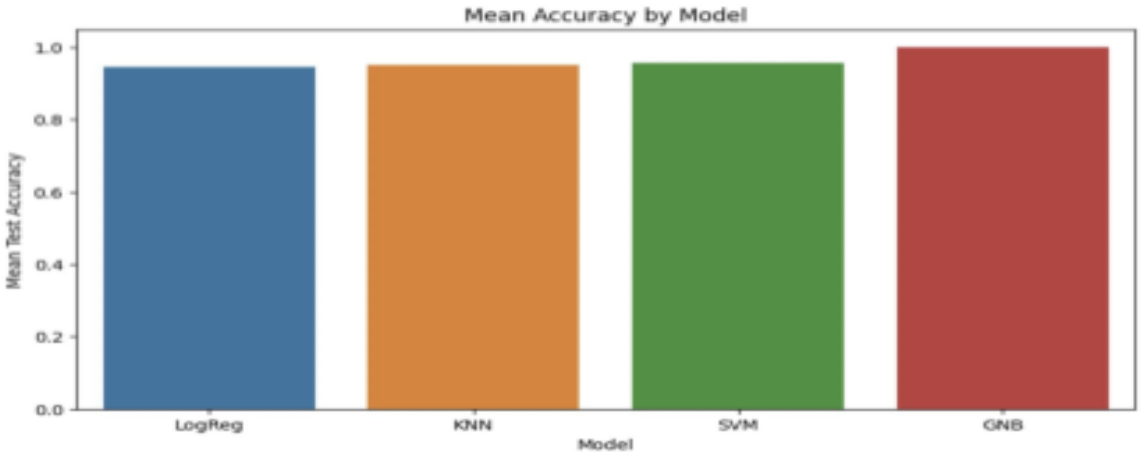
	precision	recall	f1-score	support
malignant	0.95	0.95	0.95	7673
benign	0.95	0.95	0.95	7624
accuracy			0.95	15297
macro avg	0.95	0.95	0.95	15297
weighted avg	0.95	0.95	0.95	15297

SVM

	precision	recall	f1-score	support
malignant	0.95	0.96	0.96	7673
benign	0.96	0.95	0.95	7624
accuracy			0.95	15297
macro avg	0.95	0.95	0.95	15297
weighted avg	0.95	0.95	0.95	15297

GNB

	precision	recall	f1-score	support
malignant	1.00	1.00	1.00	7673
benign	1.00	1.00	1.00	7624
accuracy			1.00	15297
macro avg	1.00	1.00	1.00	15297
weighted avg	1.00	1.00	1.00	15297



- **Host based Cryptojacking**

LogReg

	precision	recall	f1-score	support
malignant	0.99	0.99	0.99	31333
benign	0.99	0.99	0.99	31208
accuracy			0.99	62541
macro avg	0.99	0.99	0.99	62541
weighted avg	0.99	0.99	0.99	62541

KNN

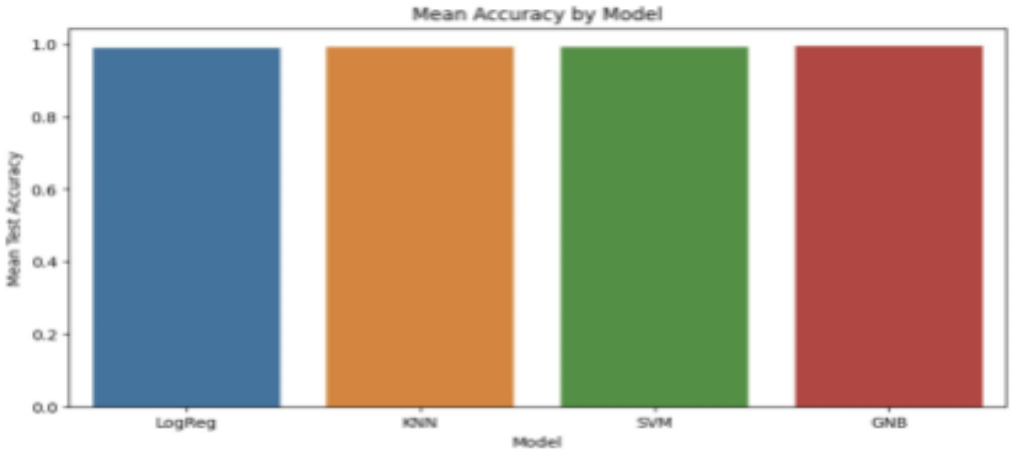
	precision	recall	f1-score	support
malignant	0.99	0.99	0.99	31333
benign	0.99	0.99	0.99	31208
accuracy			0.99	62541
macro avg	0.99	0.99	0.99	62541
weighted avg	0.99	0.99	0.99	62541

SVM

	precision	recall	f1-score	support
malignant	0.99	0.99	0.99	31333
benign	0.99	0.99	0.99	31208
accuracy			0.99	62541
macro avg	0.99	0.99	0.99	62541
weighted avg	0.99	0.99	0.99	62541

GNB

	precision	recall	f1-score	support
malignant	1.00	0.99	0.99	31333
benign	0.99	1.00	0.99	31208
accuracy			0.99	62541
macro avg	0.99	0.99	0.99	62541
weighted avg	0.99	0.99	0.99	62541



4. Scenario 4:

In this experimental scenario, we assumed that the attacker(s) targeted all devices within the smart home environment, resembling a scenario where various network-based attacks are employed. To conduct the experiment, we utilized the entire dataset available, incorporating data from all devices within the smart home network.

LogReg

	precision	recall	f1-score	support
malignant	0.98	0.97	0.97	38943
benign	0.97	0.98	0.97	38911
accuracy			0.97	77854
macro avg	0.97	0.97	0.97	77854
weighted avg	0.97	0.97	0.97	77854

KNN

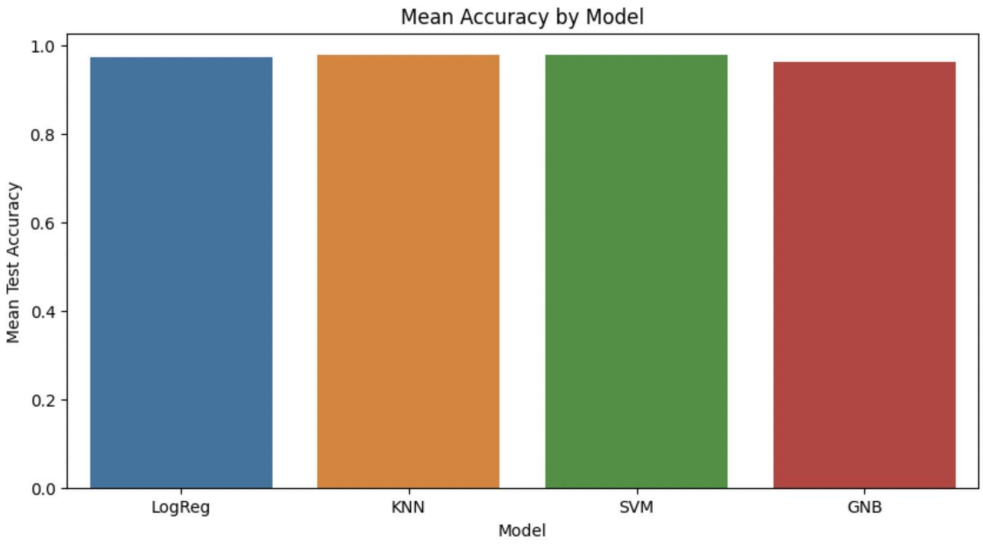
	precision	recall	f1-score	support
malignant	0.98	0.98	0.98	38943
benign	0.98	0.98	0.98	38911
accuracy			0.98	77854
macro avg	0.98	0.98	0.98	77854
weighted avg	0.98	0.98	0.98	77854

SVM

	precision	recall	f1-score	support
malignant	0.99	0.97	0.98	38943
benign	0.97	0.99	0.98	38911
accuracy			0.98	77854
macro avg	0.98	0.98	0.98	77854
weighted avg	0.98	0.98	0.98	77854

GNB

	precision	recall	f1-score	support
malignant	0.97	0.95	0.96	38943
benign	0.95	0.97	0.96	38911
accuracy			0.96	77854
macro avg	0.96	0.96	0.96	77854
weighted avg	0.96	0.96	0.96	77854

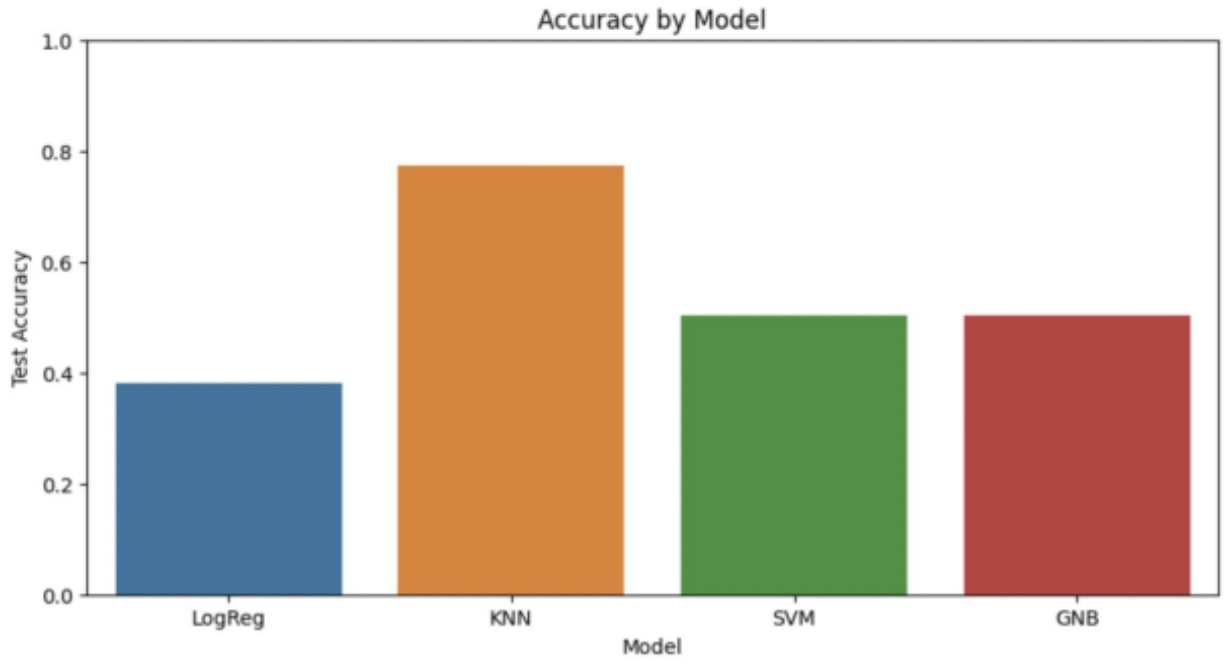


We conducted several experiments to evaluate the performance of our classifier in different scenarios, and we observed that it consistently achieved high accuracy in detecting cryptojacking attacks. However, it is worth noting that in our experiments, the GNB classifier outperformed the other classifiers, which contrasts with the findings reported in the paper where the SVM classifier was deemed the most effective. This difference in results suggests that when conducting a comprehensive set of experiments similar to those described in the paper, the SVM classifier may demonstrate superior performance. Therefore, in our future experiments focusing on attacks, we are primarily comparing the results obtained using the SVM classifier. Subsequently, we launched the aforementioned attacks on our model to assess its accuracy, and the following results were obtained.

Introducing dummy packets: By including additional packets containing irrelevant or dummy data into the dataset used for training the ML model, we noted a significant decrease in accuracy. Specifically, the previously highly accurate model, which achieved 99% accuracy with the original datasets, experienced a substantial drop to 50% accuracy when utilizing the SVM classifier.

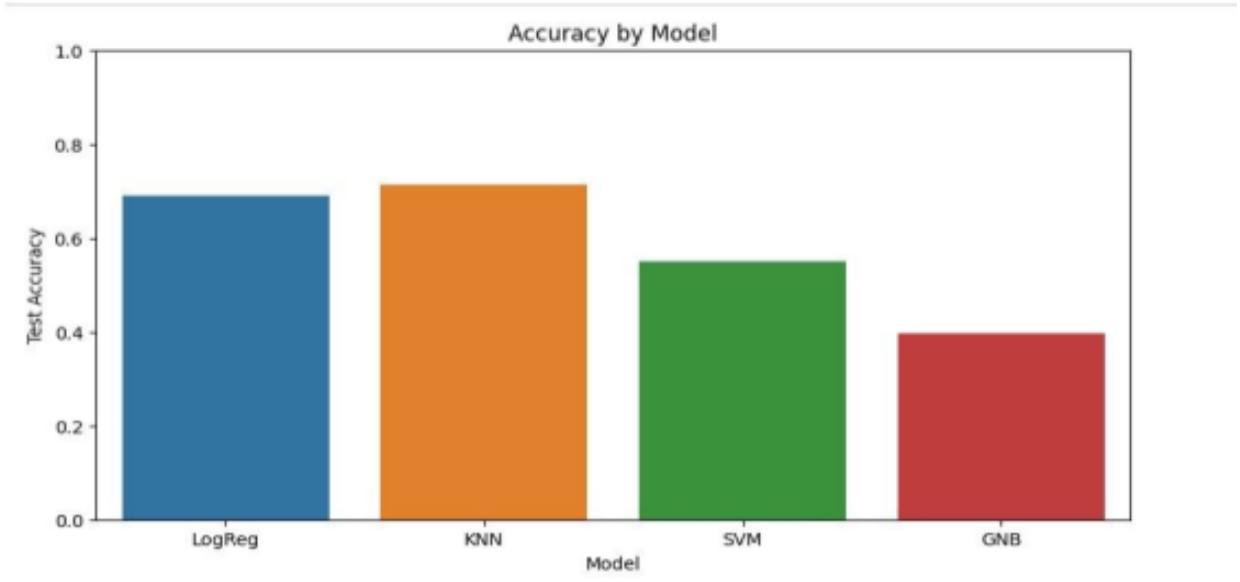
This suggests that the introduction of these irrelevant packets had a detrimental impact on the model's performance.


```
LogReg
{'malignant': {'precision': 0.30392684238838086, 'recall': 0.1805111821086262, 'f1-score': 0.22649829625175388, 'supp
ort': 6260}, 'benign': {'precision': 0.4122364802933089, 'recall': 0.5816359521500162, 'f1-score': 0.482499664744535
4, 'support': 6186}, 'accuracy': 0.37988108629278483, 'macro avg': {'precision': 0.35808166134084485, 'recall': 0.381
07356712932117, 'f1-score': 0.35449898049814466, 'support': 12446}, 'weighted avg': {'precision': 0.3577596738265847,
'recall': 0.37988108629278483, 'f1-score': 0.3537379287036538, 'support': 12446}}
KNN
{'malignant': {'precision': 0.7395075545607163, 'recall': 0.844408945686901, 'f1-score': 0.7884844868735084, 'suppor
t': 6260}, 'benign': {'precision': 0.816157040392601, 'recall': 0.6989977368250889, 'f1-score': 0.7530477185649599,
'support': 6186}, 'accuracy': 0.7721356259039048, 'macro avg': {'precision': 0.7778322974766586, 'recall': 0.77170334
12559949, 'f1-score': 0.7707661027192341, 'support': 12446}, 'weighted avg': {'precision': 0.7776044306137484, 'recal
l': 0.7721356259039048, 'f1-score': 0.7708714506565165, 'support': 12446}}
SVM
{'malignant': {'precision': 0.5029728426803792, 'recall': 1.0, 'f1-score': 0.6693039666417192, 'support': 6260}, 'ben
ign': {'precision': 0.0, 'recall': 0.0, 'f1-score': 0.0, 'support': 6186}, 'accuracy': 0.5029728426803792, 'macro av
g': {'precision': 0.2514864213401896, 'recall': 0.5, 'f1-score': 0.3346519833208596, 'support': 12446}, 'weighted av
g': {'precision': 0.2529816804739815, 'recall': 0.5029728426803792, 'f1-score': 0.33664171871903925, 'support': 1244
6}}
GNB
{'malignant': {'precision': 0.5031345442854847, 'recall': 1.0, 'f1-score': 0.6694471179552989, 'support': 6260}, 'ben
ign': {'precision': 1.0, 'recall': 0.0006466214031684449, 'f1-score': 0.0012924071082390954, 'support': 6186}, 'accur
acy': 0.5032942310782581, 'macro avg': {'precision': 0.7515672721427423, 'recall': 0.5003233107015842, 'f1-score': 0.
335369762531769, 'support': 12446}, 'weighted avg': {'precision': 0.750090169309588, 'recall': 0.5032942310782581, 'f
1-score': 0.3373560813732716, 'support': 12446}}
```

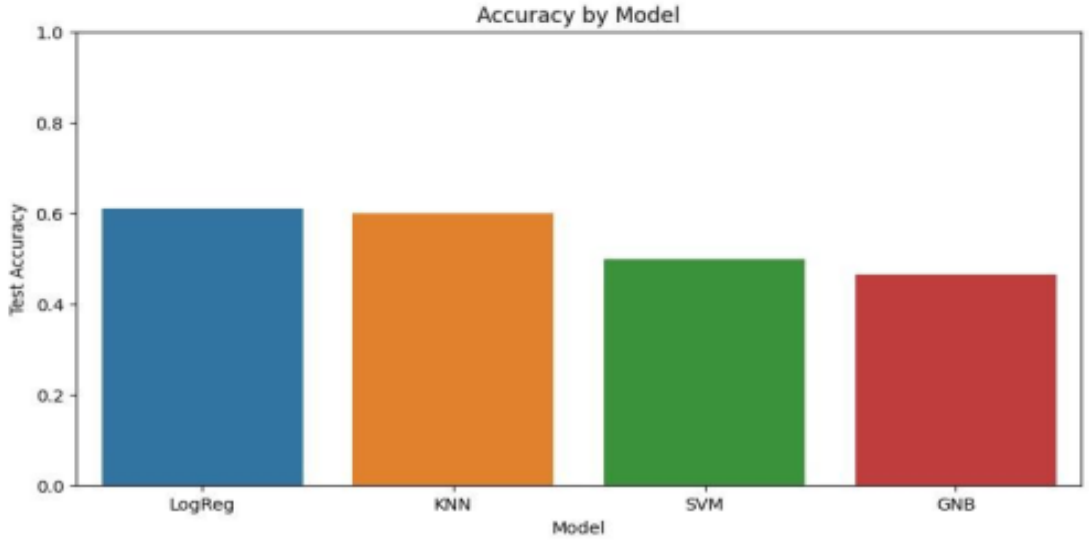
Padding: When employing the perturbation method of packet padding, which involves adding extra zero bytes to adjust the packet size and modify the length and checksum fields, the accuracy of the detection mechanism utilizing SVM decreased to 53%. This reduction in accuracy suggests that the detection mechanism's performance was noticeably impacted when confronted with this particular perturbation.

```
Feature Extraction: 100%|██████████| 6055/6055 [03:51<00:00, 26.15it/s]
WARNING:tsfresh.feature_extraction.settings:Dependency not available for matrix_profile, this feature will be disabl
ed!
Feature Extraction: 100%|██████████| 6260/6260 [04:22<00:00, 23.86it/s]
WARNING:tsfresh.feature_extraction.settings:Dependency not available for matrix_profile, this feature will be disabl
ed!
Feature Extraction: 100%|██████████| 6055/6055 [03:52<00:00, 26.06it/s]
relevance_table  (255, 4)
relevance_table_changed_mal  (258, 4)
let the ml starts
LogReg
{'malignant': {'precision': 0.6409164172231177, 'recall': 0.889297124600639, 'f1-score': 0.7449484811989829, 'suppor
t': 6260}, 'benign': {'precision': 0.8090383025626895, 'recall': 0.48488852188274156, 'f1-score': 0.606361007847996
8, 'support': 6055}, 'accuracy': 0.6904587900933821, 'macro avg': {'precision': 0.7249773598929036, 'recall': 0.6870
928232416903, 'f1-score': 0.67565474452349, 'support': 12315}, 'weighted avg': {'precision': 0.7235780506564192, 're
call': 0.6904587900933821, 'f1-score': 0.6768082334409463, 'support': 12315}}
KNN
{'malignant': {'precision': 0.6987319632706602, 'recall': 0.765814696485623, 'f1-score': 0.7307369865101744, 'suppor
t': 6260}, 'benign': {'precision': 0.7312064539787312, 'recall': 0.6586292320396366, 'f1-score': 0.6930228516812929,
'support': 6055}, 'accuracy': 0.7131140885099472, 'macro avg': {'precision': 0.7149692086246957, 'recall': 0.7122219
642626297, 'f1-score': 0.7118799190957337, 'support': 12315}, 'weighted avg': {'precision': 0.7146989174921276, 'rec
all': 0.7131140885099472, 'f1-score': 0.7121938207457508, 'support': 12315}}
SVM
{'malignant': {'precision': 0.5316937997921718, 'recall': 0.9808306709265175, 'f1-score': 0.6895777178796046, 'suppo
rt': 6260}, 'benign': {'precision': 0.8435462842242504, 'recall': 0.10685383980181667, 'f1-score': 0.189680445617121
08, 'support': 6055}, 'accuracy': 0.5511165245635404, 'macro avg': {'precision': 0.687620042008211, 'recall': 0.5438
422553641671, 'f1-score': 0.4396290817483628 5, 'support': 12315}, 'weighted avg': {'precision': 0.6850244366769656,
'recall': 0.5511165245635404, 'f1-score': 0.4437898182816073, 'support': 12315}}
GNB
{'malignant': {'precision': 0.44323377760482535, 'recall': 0.7277955271565495, 'f1-score': 0.5509402019469133, 'supp
ort': 6260}, 'benign': {'precision': 0.16306483300589392, 'recall': 0.05483071841453344, 'f1-score': 0.0820664936349
0298, 'support': 6055}, 'accuracy': 0.3969143321153065, 'macro avg': {'precision': 0.30314930530535966, 'recall': 0.
3913131227855415, 'f1-score': 0.31650334779090816, 'support': 12315}, 'weighted avg': {'precision': 0.30548120273299
995, 'recall': 0.3969143321153065, 'f1-score': 0.3204058695206671, 'support': 12315}}
```



Splitting: When applying the perturbation technique of packet splitting, which involves breaking a packet into multiple fragments, the accuracy of the detection mechanism utilizing SVM significantly dropped to 47%. This indicates that the effectiveness of the detection mechanism was greatly compromised when faced with this specific perturbation.

```
0 NAN in malicious
0 NAN in benign
Feature Extraction: 100%|██████████| 6055/6055 [04:00<00:00, 25.16it/s]
WARNING:tsfresh.feature_extraction.settings:Dependency not available for matrix_profile, this feature will be disabl
ed!
Feature Extraction: 100%|██████████| 6260/6260 [03:58<00:00, 26.29it/s]
WARNING:tsfresh.feature_extraction.settings:Dependency not available for matrix_profile, this feature will be disabl
ed!
Feature Extraction: 100%|██████████| 6305/6305 [04:25<00:00, 23.72it/s]
relevance_table  (255, 4)
relevance_table_changed_mal  (258, 4)
let the ml starts
LogReg
{'malignant': {'precision': 0.5637062019861345, 'recall': 0.9611821086261981, 'f1-score': 0.7106413133341207, 'suppo
rt': 6260}, 'benign': {'precision': 0.8714965626652564, 'recall': 0.26137985725614593, 'f1-score': 0.402147388970229
4, 'support': 6305}, 'accuracy': 0.6100278551532033, 'macro avg': {'precision': 0.7176013823256955, 'recall': 0.6112
809829411721, 'f1-score': 0.5563943511521751, 'support': 12565}, 'weighted avg': {'precision': 0.7181525389604173, '
recall': 0.6100278551532033, 'f1-score': 0.5558419346541099, 'support': 12565}}
KNN
{'malignant': {'precision': 0.5941673062164237, 'recall': 0.618370607028754, 'f1-score': 0.606027397260274, 'support
': 6260}, 'benign': {'precision': 0.6051239669421488, 'recall': 0.5806502775574941, 'f1-score': 0.5926345609065156,
'support': 6305}, 'accuracy': 0.5994428969359331, 'macro avg': {'precision': 0.5996456365792862, 'recall': 0.5995104
422931241, 'f1-score': 0.5993309790833947, 'support': 12565}, 'weighted avg': {'precision': 0.599665256544772, 'reca
ll': 0.5994428969359331, 'f1-score': 0.5993069966864223, 'support': 12565}}
SVM
{'malignant': {'precision': 0.4982093115797851, 'recall': 1.0, 'f1-score': 0.6650730411686586, 'support': 6260}, 'be
nign': {'precision': 0.0, 'recall': 0.0, 'f1-score': 0.0, 'support': 6305}, 'accuracy': 0.4982093115797851, 'macro a
vg': {'precision': 0.24910465578989255, 'recall': 0.5, 'f1-score': 0.3325365205843293, 'support': 12565}, 'weighted
avg': {'precision': 0.24821251814480338, 'recall': 0.4982093115797851, 'f1-score': 0.33134558199091146, 'support': 1
2565}}
GNB
{'malignant': {'precision': 0.4749321077919365, 'recall': 0.7263578274760384, 'f1-score': 0.5743337122647467, 'suppo
rt': 6260}, 'benign': {'precision': 0.4272818455366098, 'recall': 0.20269627279936558, 'f1-score': 0.274956970740103
3, 'support': 6305}, 'accuracy': 0.46358933545563075, 'macro avg': {'precision': 0.45110697666427313, 'recall': 0.46
4527050137702, 'f1-score': 0.424645341502425, 'support': 12565}, 'weighted avg': {'precision': 0.45102164989143234,
'recall': 0.46358933545563075, 'f1-score': 0.42410925103809516, 'support': 12565}}
```



Limitation:

The detection mechanism heavily relies on packet features such as packet length and timestamp, which may not be reliable for accurate detection. The evaluation was conducted with limited device data and may yield different results on a larger scale. Additionally, the mechanism can only identify already affected devices and does not provide preventive measures against malware attacks.

Conclusion and Future work:

Our evaluation revealed that various packet manipulations, including packet splitting, dummy packet insertion, and the use of proxy networks, can effectively evade the detection capabilities of the model mentioned above. As a future direction, we propose the development of countermeasures to enhance the robustness of the detection mechanism against perturbations. Several possible countermeasures can be considered

Outlier detection: Applying outlier detection techniques can help identify and remove anomalous packets that are likely to have been perturbed. This can reduce false positives and enhance the accuracy of the detection mechanism.

Protocol verification: Utilizing protocol verification techniques can ensure that packets adhere to expected protocol specifications. Inconsistent packets, indicating potential perturbations, can be detected and discarded.

Traffic normalization: Implementing traffic normalization techniques can standardize packet features such as length and inter-arrival time. This normalization can mitigate the impact of perturbations on the detection mechanism.

Encryption and authentication: Employing encryption and authentication methods can safeguard network traffic against tampering and ensure packet authenticity. This prevents the introduction of packet perturbations into the network.

Machine learning-based techniques: Utilizing machine learning approaches like adversarial training or model ensembling can enhance the detection mechanism's robustness against perturbations. Training the model on perturbed data can increase its resilience to adversarial attacks.

References

1) “[2103.03851] SoK: Cryptojacking Malware.” 2021. arXiv. https://arxiv.org/abs/2103.03851.
2) “A Foreground-Aware Framework for Local Face Attribute Transfer.” 2021. PubMed. https://pubmed.ncbi.nlm.nih.gov/34065640/.
3) “A Lightweight IoT Cryptojacking Detection Mechanism in Heterogeneous Smart Home Networks - NDSS Symposium.” n.d. Network and Distributed System Security (NDSS) Symposium. Accessed May 13, 2023. https://www.ndss-symposium.org/ndss-paper/auto-draft-196/.
4) “turkmia.” n.d. ″ - Wiktionary. Accessed May 13, 2023. https://turkmia.net/TurkMIA2023-Proceedings.pdf.