Malware's Polymorphism Analysis using a Hybrid Machine Learning Algorithm Approach

Halilou Claude Bobo Hamadjida[a], Aurelle Tchagna Kouanou[a,b], Christian Tchapga Tchito[b], Clarence Tamko Kouadjo[a]

[a]*Department of Training, Research, Development and Innovation, InchTech's Solutions, Yaounde, Cameroon, Yaoundé, PO Box 30109, Cameroon*
[b]*Department of Computer Engineering, College of Technology, University of Buea, Buea, PO Box 63, Cameroon*

Abstract

Background and Objective: The emergence of new technologies, such as artificial intelligence (AI), continues to enhance human life and activities. With advancements in information technology, communication, robotics, and Industrial Control Systems (ICS), accessing and utilizing powerful computational resources has become increasingly feasible. However, these same technologies can also aid malware in bypassing modern cybersecurity defenses by enhancing its capabilities. Polymorphism is a key example of an advanced malware technique that can be exploited using machine learning (ML). This research aims to address malware polymorphism by leveraging hybrid machine learning (HML) approaches.

Method: Building on insights from previous research, this study focuses on selecting an appropriate dataset and HML algorithm to achieve high-precision polymorphism detection. The objective is not only to detect polymorphic malware with high accuracy but also to provide real-time descriptions of malware tactics, techniques, and procedures (TTPs) to improve decision-making in cybersecurity. To implement this approach, the CIC2022 malware dataset from the Canadian Institute of Cybersecurity was cleaned, formatted, pre-processed, and trained using an HML algorithm that combines Fuzzy Ranking (FR) and Support Vector Machine (SVM). The performance of the proposed method was evaluated using a confusion matrix, cross-validation AUC-ROC curve, F1 score, and false positive rate (FPR). Finally, the trained model was tested on simulated polymorphic malware to analyze its actual TTPs.

Results: The integration of SVM with the selection of FR-based characteristics achieved an overall precision rate that exceeded 0.95 for polymorphism detection. Furthermore, using the CIC2022 dataset, the model provided an approximate description of malware TTPs, achieving an even higher accuracy (0.9977) in the Microsoft BiG 2015 dataset when tested within an isolated Windows environment.

Conclusion: The proposed approach demonstrates stability, efficiency, and reliability in detecting polymorphic malware. However, there was a slight deviation from the original research hypothesis regarding the dataset used, as CIC2022 was chosen over Malimg due to accessibility constraints.

Keywords: Malware polymorphism, Tactics Techniques Procedures, Hybrid Machine Learning, Malware Classification, Cybersecurity datasets.

---

1. Introduction

Human life is continually improving. Thanks to the latest technological trends, easing health, transportation, communication, or even electricity is just becoming a matter of time. While humanity is facing a critical period of its history, the dependency on Information Technology (IT) has to be highlighted and regulated as fast as possible. As is well known, there is no perfect system in the world, nor a system completely out of danger for users and customers. This is the reason why there is no exhaustion of resources provided by enterprises in order to reinforce the quality of their products, services, and goods. Cybersecurity concerns are also one of the most prominent subjects that should be brought to this scale, because of the threat landscape which is becoming unpredictable and uncontrollable. Malware, as one of the main cybersecurity priorities, can be defined as a common type of cyberattack in the form of malicious software. Families of malware comprise cryptominers, viruses, ransomware, worms and spyware. Its common objectives are and not limited to information or identity theft, espionage, and service disruption [1]. Latest next-generation trend of technology is Artificial Intelligence (AI), which is quite productive in terms of efficient automation, orchestration, rapidity and driven decision making. In short, AI refers to an interdisciplinary field encompassing biology, computer science, philosophy, mathematics, engineering, and robotics, and cognitive science centered on simulating human intelligence using computer-based technologies [2, 3]. Various realms are being continuously developed and enhanced under this

key technology. It is no surprise that the majority of AI patents cover multiple fields, with almost 70% AI-related inventions including a combination of different AI techniques or functional applications (like planning/scheduling, computer vision, among others). Telecommunications (15%), transportation (15%), life and medical sciences (12%), personal devices, computing, and human-computer interaction (11%), are the top industries that patent heavily in AI [3]. The sub-field of AI, ML can be divided into two primary families of ML algorithms: supervised and unsupervised learning. The former refers to the process of learning an unknown function using labeled training data and example input-output pairs. In contrast, unsupervised learning refers to the detection of previously unnoticed patterns and information in an unlabeled data set [3]. These technological advances offer both promise and challenge, transforming the way cybersecurity defenses are strengthened and introducing novel threats that demand attention [2]. The predicted trends in cybersecurity and their implication has been detailed in table 2.

Table 1: Predicted trends in Cybersecurity [2]

Trends	**Emerging Technologies**	**Roles**	**Implications**
Quantum Computing	Unprecedented processing power	Enhancing Cybersecurity defenses	Potential to break current encryption methods. Necessitates the development of quantum-resistant encryption algorithms. Urgent need for proactive response to safeguard data and information integrity.
Artificial Intelligence (AI)	Real-time threat detection	Posing new threats	Enables real-time threat detection and response. Empowers proactive incident response. Enhanced accuracy and speed in identifying potential attacks. AI can be exploited for sophisticated cyberattacks.

The capabilities of AI can be used for good or bad reasons. Further-

more, imagine a threat actor or hacker, aiming to wreak havoc on a corporate infrastructure with malicious software. Based on what has been said previously, it can easily help him realize his dream. AI can be helpful for malware camouflage techniques (encryption, oligiomorphism, polymorphic, and metamorphism) and obfuscation techniques (code injection, instruction replacement, subroutine reordering, register re-assignment) [4]. By the way, for the special case of polymorphism, viruses alter their appearance by using various obfuscation techniques. One of the most famous uses of this technique was for the WannaCry case in 2017, which has encrypted more than 200,000 computers in 150 countries to demand a ransom. As the threat landscape is gradually evolving, it is extremely important to think about enhancing security facilities and processes. Based on the actual scenario, prominent attention should be paid, from some point of view, comprising:

- Increasing malware mutation and complexity;
- Lack of efficiency provided by a single machine learning algorithm structure;
- Malware description mechanism becoming useless and inaccurate against complex malware structure.

Always in the sake of enhancing the latest Cybersecurity capabilities and processes, the present research activity aims to tackle Malware's polymorphism by leveraging hybrid machine learning approaches. It also answer to the question of the affordability and the scalability of multiple ML algorithm for a probable combination or Research question Number 1(RQ1); the procedure and the resources to reach an affordable precision rate on polymorphism detection(RQ2); and at last, the way of providing as detailed as possible real-time malware's description to enhance decision taking for their combat(RQ3). Based on pre-research knowledge, let us consider the main hypothesis that FR-SVM trained on dataset and combined with Mitre (Tactics Techniques Procedures)TTPs yields an affordable precision rate for malware polymorphism detection(HP). Moreover, HP1: All ML algo. type Can be combined for this research sake; HP2: Supervised ML algorithm Training process and considerable dataset are sufficient to reach an affordable precision rate; and HP3: Mitre attack framework is enough for retrieving and describing a malware ttps can consider as the research trajectory. After in-depth deep research has been performed, developing a concrete approach will

be the major concern of this paper. The remainder of this paper is organized as follows. Section 2 discusses related work and is followed by a discussion of the malware detection and analysis mechanism using AI. The methodology of the present approach is proposed in Section 3. The results and discussions are presented in Section 4 and Section 5 concludes the paper with a summary and future research directions. Literally, every domain related to cyberspace is targeted by this research. Some of them are:

- Cybersecurity and Cyber war;
- Industry (Industrial Control System or ICS);
- Army;
- Transport and Trade.

2. Related Works

In order to carry out this analysis, some research was carried out on what has already been done in the concept of malware detection and analysis mechanism using AI. This review of the literature is implicitly divided into three segments. The first segment is concerning researches made in latest ML approaches for archiving considerable precision rate in malware analysis; the second segment discussed on dataset sources used for archiving acceptable performances; and the last segment is reviewing attack Tactics Techniques and Procedures (TTPs) platforms available providing a comprehensive malware's description. In this section, a brief overview is given on authors' surveys, novelties, findings and results. In 2024 Wang, Y. *et al.* [5], proposed a novel Deep Learning (DL) Based Malware Attack DetectoR in Android Smartphones using LinkNET(MADRAS-NET) which effectively detects and mitigates the types of malwares in Android devices. That paper proposed MADRAS-NET technique is validated by using the Cloud Simulator (Cloudsim). Furthermore, an AndMal2020 dataset, which includes 400,000 Android apps and contains 200,000 benign malware samples as well as 200,000 samples belonging to 14 key malware categories and 191 important malware families, is tested in this technique which accurately classifies the majority of the occurrences of malware categories, and malware families and demonstrates its efficacy.
In 2024 Maniriho, P. *et al.* [6], presented MeMalDet, a novel memory

analysis-based malware detection technique using deep autoencoders and stacked ensemble learning. MeMalDet extracts optimal features from memory dumps using deep autoencoders in an unsupervised manner, avoiding manual feature engineering. A stacked ensemble of supervised classifiers then performs highly accurate malware detection. The improved dataset(MemMal-D2024, a dataset has 58 features (attributes) with a total number of 58,596 records (29,298 malware and 29,298 benign) extracted from memory images captured during memory analysis. enables temporally robust evaluations, which is a novel contribution. In May 2024 Fleming, M. *et al.* [7], performed a study, which sought to determine whether or not fuzzy hashes are always effective, how quickly malware is evolving, and how malware evolution affects fuzzy hashing. Experiments with known malware family and analysis with over 4500 APK files, including 100 benign samples collected from 2012 - 2022 were conducted using various fuzzy hashing algorithms(from virusShare and Virustotal), file-level and section-level similarity hashing.

In 2024 [8], to ascertain the efficacy of the FSVM model, researchers employed a publicly available dataset from Kaggle, which encompasses two distinct decision labels. The proposed evaluation methodology involves a comprehensive comparison of the classification accuracy of the processed dataset against four contemporary models in the field. This latest research proposal concluded that up to 3% of accuracy can be enhance by implementing that method. In 2023, Hoang Hai *et al.* [9], performed research aiming to integrate EDR with an image-based malware classifier. A basic EDR implementation named Deep Ocean Protection System (DOPS) has been developed with two pre-trained models (Mobilenet V2 and Inception V3) fine-tuned with MalImg and BODMAS datasets. The models were evaluated with the DikeDataset and Mobilenet V2 fine-tuned with BODMAS 4.0.0 performed best in terms of loading and prediction time with a high AUC score of 0.8615. Inception V3 fine-tuned with BODMAS 4.0.0 also achieved a remarkable AUC score of 0.9392. These results show the potential of integrating image-based malware detection with EDR. In 2023, Khan *et al.* [10], employs a Generative Adversarial Network (GAN) based Malware Classifier Optimizer (MCOGAN) framework, which can optimize a malware classifier. This framework utilizes GANs to generate fabricated benign files that can be used to train external discriminators for optimization purposes.

In 2022 Manoj Kumar *et al.* [11], in order to tackle a malware variant detection model that combines different behavioral activities, proposed a Deep-Ensemble and Multifaceted Behavioral Malware Variant Detection Model

using Sequential Deep Learning and Extreme Gradient Boosting Techniques. Different behavioral features were extracted from the dynamic analysis environment. Then, a feature extraction algorithm that can automatically extract effective representative patterns has been designed and developed to extract the hidden representative features of the malware variants using google translate a sequential deep learning model. The dataset utilized in this study has 23070 samples, with 19076 malware samples and 3994 benign ones(from the Vx Heaven public repository).

In 2021, Alan *et al.* [12], presented a novel deep-learning-based architecture which can classify malware variants based on a hybrid model. The main contribution of the study is to propose a new hybrid architecture which integrates two wide-ranging pre-trained network models in an optimized manner.he proposed method tested on Malimg, Microsoft BIG 2015, and Malevis datasets. The experimental results show that the suggested method can effectively classify malware with high accuracy which outperforms the state of the art methods in the literature. In 2020 Cordeiro de Amorim *et al.* [13], proposed an iterative data pre-processing method capable of aiding to increase the separation between clusters. It does so by calculating the within-cluster degree of relevance of each feature, and then it uses these as a data rescaling factor. By repeating this until convergence our malware data was separated in clear clusters, leading to a higher average silhouette width.

In 2019 Tong *et al.* [14], proposed a novel malware detection approach based on the family graph. First, API calls of the monitored application are traced, and then the dependency graph based on the dependency relationship of the API calls is generated. At last, the family dependency graph via clustering the graphs of a known malware family is constructed. In this way, it can determine whether a new sample belongs to a known malware family. For this research, a malware dataset obtained from Anubis(malware set of 300 samples, benign set of 3546 samples) was used. In 2014 Bai *et al.* [15], proposed a malware detection approach by mining format information of PE (portable executable) files. Based on in-depth analysis of the static format information of the PE files, 197 features were extracted from format information of PE files and applied feature selection methods to reduce the dimensionality of the features and achieve acceptable high performance.

A key element of previous research activities is malware dataset, which presents information of known malware signatures and features. In 2024, Smmarwar *et al.* [16], performed a compressible review on malware detection and identification framework by leveraging ML and DL. Some dataset

enumerated include:

- CICMalMem2022: dataset is a obfuscated malware for memory analysis. It is created to test obfuscated malware detection methods through memory dumps. It is consists of 58,596 instances, in which 29,298 instances are benign class and 29,298 are malware class.

- Malimg: windows malware dataset that contains 9339 grayscale malware images of windows executable files that belong to 25 families of malware such as worms, Trojans, PWS, dialer, Downloader, rogue, Backdoor, and Worm:autoIT. The malware binaries of 8-bit are transformed into grayscale images.

- Microsoft Malware classification Challenge BIG 2015 (MMB-15): or MMB-15, is more than half a terabyte in size. It is a collection of 9 types of malware families. Each malware file is distinguished by an identifier, a twenty-character unique hash value and a class label which is separating each of the 9 malware family names. A total of 10,349 malware samples are collected in this work of worms, adware, backdoor, Trojan and obfuscated malware attacks.

Malware description is mostly related to the concept of TTPs. While a dataset is implicit and more technical, Threat intelligence sources give more explanation of a malware capabilities hit-her-to the Advanced Persistent Threat Group(APT Group) author of a mischievous activity. Some of them include [17]:

- Mitre Attack: is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies in the private sector, in government, and in the cybersecurity product and service community.

- OSINT Fromework: OSINT framework focused on gathering information from free tools or resources. The intention is to help people find free Open Source Intelligence (OSINT) resources.

- VirusTotal: It is publicly available website for collecting malware samples and benign samples.

- OpenCTI: OpenCTI is an open source platform allowing organizations to manage their cyber threat intelligence knowledge and observables. It has been created in order to structure, store, organize and visualize technical and non-technical information about cyber threats.

Based on the related works, having a malware dataset([10,349-58,596] lines and [9-179] malware families), a joint ML/DL algorithm set, at least a isolated computing environment(with basic code running library) and an open-source and well-elaborated Threat Intelligence Platform can be helpful for achieving a greater than 96.0% score for accuracy, F1-Score and True Positive Rate(TPR); a lesser than 2% for FNR; an about 0.90% for AUC-ROC and a qualitative description of a malware. The proposed approach is based on two main algorithm, Fuzzy Rank (FR) for data ordering and elimination of useless data and, Support Vector Machine (SVM) for data training and malware classification throughout the project pipeline.

3. Proposed Method

The proposed approach is a foot forward to give better understanding of this research scope and objectives. Since malware polymorphism is an important issue, collecting and using available resources and materials for their combat is all about Cybersecurity in practice. Literature reviews, AI and Cyber security advancement tailored the proposed method of this research activity summarized in the Fig. 1. The Canadian Institute of Cybersecurity(CIC) malware dataset released in 2020, is first cleaned and prepared to become "x". FR algorithm is applied on "x" for malware data classification and families building based on similarity ratio between each data. Then, "x" is analyzed to represent outlines, pull stars, dataset configurations et inter-correlations. The result "y" with 12 selected features from "x" is trained using SVM(default parameter, C=100 and C=1000). The result data "z" is evaluated with classification accuracy and error, precision, recall, f-1 score, confusion matrix and ROC-AUC. From this point, we are done with the pre-detection phase. Data A malware signature is collected in the test environment and if it is found in "z", the resulting k' signature is looked up in a publicly available malware source(Virus-Total). The resulting k"(Malware TTP) is matched using FR with precedents $k(k^{(-1)}, k^{(-2)}\dots)$ to finally print out the actual value of k. The code is essentially made with python programming language. For an objective analysis, the same process is done using the MMB 2015 dataset.

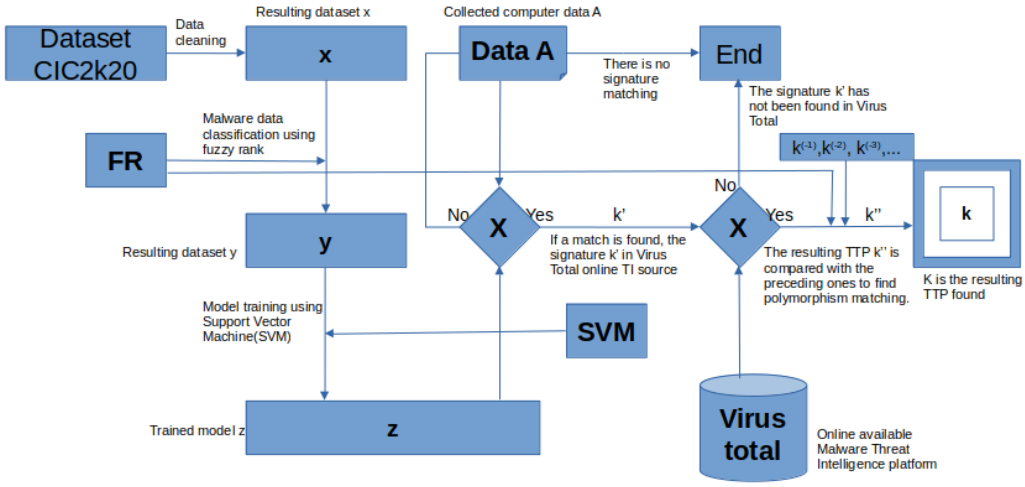

Figure 1: Proposed method's pipeline

3.1. Dataset Cleaning and Preparation

When dealing with malware in general, it is important to have a well provided source of dataset with triggered features. This is the principal source of energy of our designed pipeline. Thanks to the CIC 2020 dataset for his accessibility. There are a lot of datasets, but for this project, about 59392 lines of data samples are coming from the ransomware, the trojan and the benign dataset. Throughout the whole pipeline, a malicious signature is labeled as 1 and a benign one as 0. From "x" to "z", this consideration will remain but can be modified after the training process. This labeling is represented by a creation of the new column in the merged dataset called "Class". The resulting dataset(with the .csv extension) is now loaded to the programming environment to be manipulated in raw using the panda python framework. The dropna() function is applied on that data-frame to automatically modernize "nan" and empty rows. The resulting data "x" is ready for malware classification, data analysis, model training and the remaining step. Subsection text.

3.2. Malware family classification

In order to emphasize the issue of malware polymorphism, it is important to figure out if it is applied in the present study case "x". Multiple deep comparisons are done to determine matching patterns between each data line on "x". It is where the concept of FR comes out. The used FR approach is

based on the Levenshtein Distance(LD), which is a measure of the similarity between two strings not necessarily of the same length. The distance is the number of deletions, insert ions, or substitutions required to transform the source string a into the target string b [18]. Obviously if the strings s and t are the same then $LD(a, b) = 0$.
Lets $Ld_{a,b}(len(a), len(b))$, the Lenvenshtein distance between 'a' and 'b'.

$$Ld_{a,b}(i, j) = max(i, j), if min(i, j) = 0 \tag{1}$$

Otherwise:

$$min(Ld_{a,b}(i-1, j) + 1, Ld_{a,b}(i, j-1) + 1, Ld_{a,b}(i-1, j-1) + 1_{ai \neq bj}) \tag{2}$$

Where $1_{ai \neq bj}$ is an indicative function,

$$\begin{aligned} 1_{ai \neq bj} &= 0, a_j = b_j \\ &= 1, otherwise. \end{aligned}$$

And the $Ld_{a,b}(i, j)$ is the distance between the first i characters of 'a' and the first 'j' character of 'b'. Lets $P_{a,b}$, the following ratio:

$$\frac{Ld_{a,b}(i, j)}{max(len(a), len(b)} * 100 \tag{3}$$

Figure 2 shows the FR matching algorithm and the malware family tree building.

Although it is time consuming to evaluate the overall process on every "x" line, data results are collected for the first 100 lines and estimation is done for the remainder. In this testing scope, about malware"s signature families and sub-families is detected, or malware"s signature families for the completed samples of "x", with a sensitivity around 96%. This similarity checking also detects and removes duplicate lines or lines with a similarity ratio=100%. Gaining in precision and additional polymorphism knowledge. The resulting data-frame "y" is ready for data analysis, model training and next phases.

3.3. Data Analysis

It is judicious to have an in-depth view into the dataset characteristics to collect information on data density, distribution, weight, variation and traffic. As the proposed concept of hybrid machine learning dataset is based on

```
Begin
   Define x(dataset),
            s(sensibility)=96,
            k(classification)=[]
   d(tree)={}
   For every a,b in x:
if a or b in k:
      pass.
else
      if P_a,b>s, add (a,b) in k.
   Print(k)
   For  each (a,b) in k:
if a and b in k:
      pass.
if a in k and b not in k:
      add b as a's son({a:{b}}
if b in k and a not in k:
      add a as b's son({b:{a}}
         If a and b not in k:
      {a:{b}}
   Print(d)
End
```

Figure 2: Malware family matching and tree building

dataset, it is important to rapidly detect anomalies from the sources of information. For these reasons, Kernel Density Estimation(KDE) plot, scatter plot, and vertical pulsar is represented per chosen feature, before the model training process. KDE is a non-parametric method used for efficient visualization of probability density functions, of a random variable based on kernels or weights. For the present research activity, emphasize and estimate the occurrence rate of a data value is prominent to materialize data the general data density. Let $(x_1, x_2, \ldots, x_n)$ be identically distributed and independent samples taken from some uni-variate distribution with an unknown density f at any given point x. If the shape of the function f needs to be estimated, its kernel density estimator is:

$$\widehat{f}_n(x) = \frac{1}{n}\sum_{k=0}^{n}(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x - x_i}{h}) \tag{4}$$

Another important interest behind this data analysis process is the scatter plot, to display values for typically two variables for a set of data, so as to materialize malware data features to feature inter-corelation. So each feature couple (x, y), data intersection points are represented. The idea behind the heatmap is to interpret the used malware dataset as a well provided matrix, with the dimension number equal to the number of features(so $n = 12$). Using the representation, data density and traffic is visualized and evaluated within the feature scope. For the goal the vertical pulsar plot(or bar plot) in the evaluate and the occurrence of every data value in a given column(feature) of the malware dataset.
There are many other representations for data analysis, but for the scope of the research activity, The mentioned plots will be enough. This little footprinting marks the end of the pre-detection phase. With the information collected, the manipulated malware datasaset is ready for the supervised training procedure.

3.4. Model training and evaluation

Support Vector Machine(SVM) supervised machine learning algorithm, Effective for high-dimensional spaces and classification tasks, is fed with "y" to build the proposed model. This model is afterward trained and evaluated. For programming purposes, the skiti-learn python framework is used in order to train the model.
Consider the training set of two separate classes be represented by the set

of vectors [18, 19]: $(v_0, y_0), (v_1, y_1), ..., (v_{n1}, y_{n1})$. The data points, v_j only appear inside a scalar product. Let map the data points into an alternative higher dimensional space, called feature space, through:

$$V_i^T V_j \rightarrow < \phi(V_i), \phi(V_j) >$$

Where h, i denotes the scalar product in the feature space. The map $\phi(v_i)$ does not need to be known since it is implicitly defined by the choice of the positive definite kernel:

$$K(V_i, V_j) = < \phi(V_i), \phi(V_j) >$$

It is assumed that $K(v_i, v_j) = K(v_j, v_i)$. Examples are the Radial Basis Function(RBF) kernel:

$$K(V_i, V_j) = exp(-\frac{\|V_i - V_j\|}{2\sigma}) \tag{5}$$

For binary classification with a given choice of kernel the learning task therefore involves maximization of the Lagrangian:

$$Ld() = \sum_{i=0}^{n-1} \alpha_i - \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \alpha_i \alpha_j y_i y_j K(V_i, V_j)$$

Subject to the constraints $\sum_{i=0}^{n-1} \alpha_i y_j, \alpha \geq 0, i = 0, 1, 2, ..., n-1$. After the optimal values i have been found the decision function is given by:

$$f(x) = sign(\sum_{j=0}^{n-1} \alpha_i y_i K(x_i, v_i) + b)$$

The bias b is found from the primal constraints:

$$f(x) = -\frac{1}{2}(\max_{i:-1} \sum_{j=0}^{n-1} \alpha_j y_j K(x_i, v_i) + \min_{i:+1} \sum_{j=0}^{n-1} \alpha_i y_i K(x_i, y_i))$$

For the Karush-Kuhn-Tucker conditions (which are formulated for a minimum) we have to change Ld to Ld . Thus taking into account the constraints we have the Lagrangian [18, 19]:

$$L(\alpha) = \sum_{i=0}^{n-1} \alpha_i - \frac{1}{2} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \alpha_i \alpha_j \alpha_j y_i y_i K(v_i, v_j) - \sum_{j=0}^{n-1} \alpha_j y_j - \sum_{j=0}^{n-1} \alpha_j \lambda_j \tag{6}$$

Table 2: Trained Model metrics

Metrics	Description	Formula
Quantum Computing	Unprecedented processing power	Enhancing Cybersecurity defenses
Artificial Intelligence (AI)	Real-time threat detection	Posing new threats

From $\frac{\partial L}{\partial x} = 0$, we find:

$$-1 + \sum_{j=0}^{n-1} \alpha_j y_j K(v_k, v_j) - y_k - \lambda k = 0 \tag{7}$$

For k=0,1,...,n-1. The other Karush-Kuhn-Tucker conditions are:

$$\sum_{j=0}^{n-1} \alpha_j y_j = 0 \tag{8}$$

$$\begin{aligned} \alpha_j &\geq 0, j = 0, 1, ..., n1 \\ \lambda_j \alpha_j &= 0, j = 0, 1, ..., n1 \\ \lambda_j &\geq 0, j = 0, 1, ..., n1 \end{aligned}$$

Note that there is no condition on the Lagrange multiplier µ. By exploring (8) and (7), a new trained dataset is generated and is ready for the detection phase. Tab.2 summarizes parameters used for the model evaluation. The part 4 of this research paper contain the results of this process. This step mark the end of the pre-detection phase.

3.5. Signature correlation and malware description

To test the trained model, the Message Digest 5(MD5) hash "Data A" is compared with the MD5 signature of the proposed model. If a match is found, the resulting "k'" go through the Total Virus malware source for threat intelligence to find this particular malware TTPs. The generated TTP k" is correlated with precedent TTPs values using FR to figure out any polymorphism matching between two malware within a given interval of time. Technically, the process in part 3.2 is repeated but with a lesser sensitivity level. The tailed and refined value of "k" is the actual TTP that research aims to exhibit. Fig. 3 summarizes this process. Following this pipeline, every part of the proposed model is materialized. The part 4. is detailing all the research outputs of the overall process.

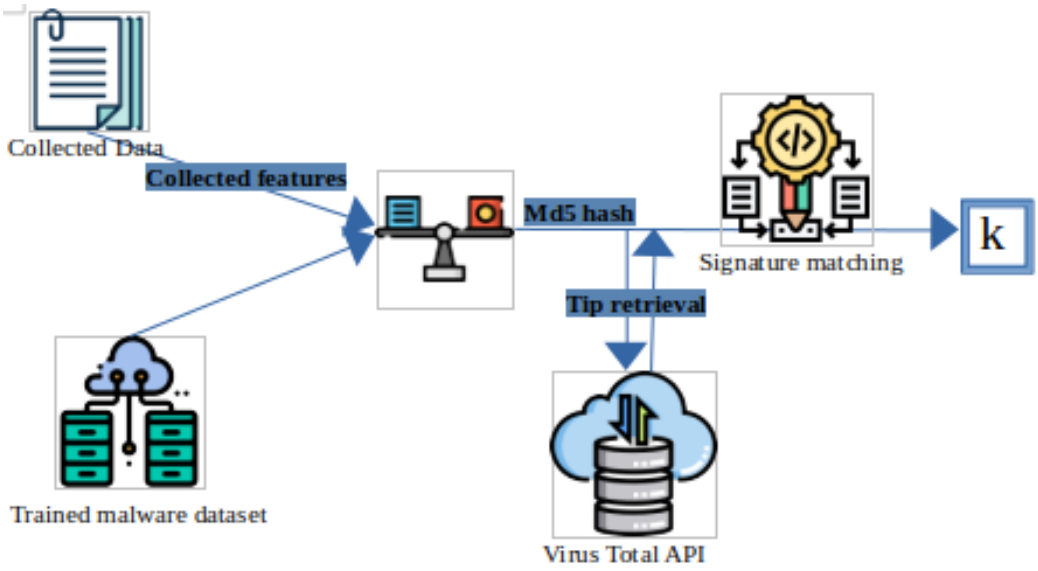

Figure 3: signature and malware correlation

4. Results and discussion

In this section, results and discussion of the proposed research method are presented in the same order than the research questions(1, 2, 3).

4.1. Results

4.1.1. Affordability and the scalability of multiple ML resources for a probable combination

Following the pipeline lamb by lamb, after FR performances, analysis data from the process of data analysis is retrieved. Abscissas and ordinary feet can vary depending on the graph and data scope but are considerably consistent and well regulated. Fig. 4 shows four different cases of vertical bar plot for exactly 9 features of the 12 selected, with a high data maximum occurrences from 5000-25000 pulsars, while minimums are actually lesser than 1000 pulsars value on the tailored dataset of this proposed research. Much malware variability can be evaluated where there are more occurrences of a given data value. Fig. 5 shows data inter-correlation of features using a multidimensional scatter plot, expressing high traffic intensity on more than average plot and extremely low density in some others(columns with 2 options(0,1)) tailored dataset of the proposed method. Traffic density is also

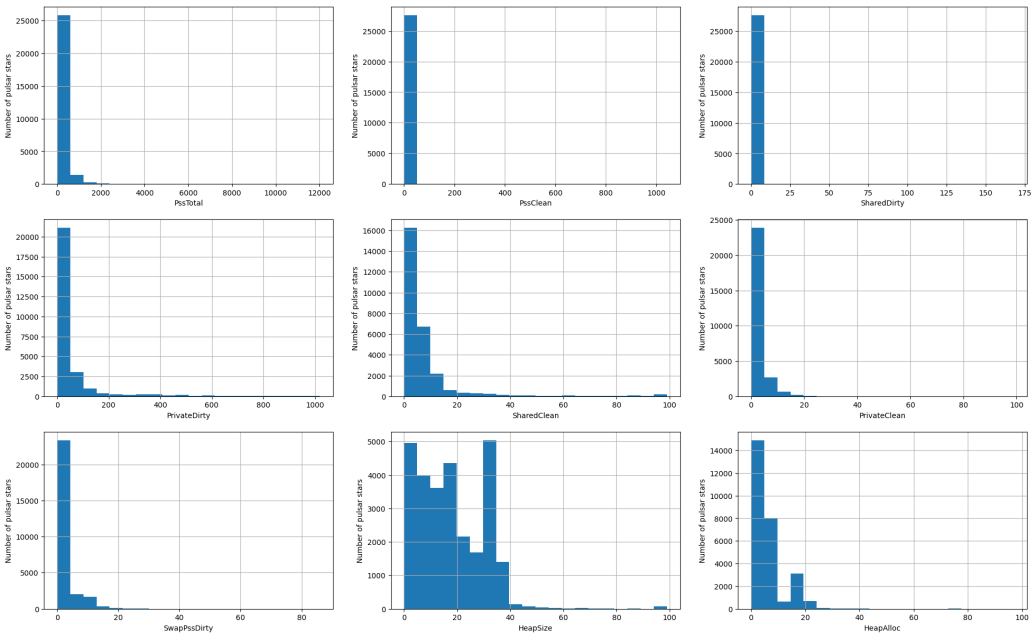

Figure 4: pulsar representation

visualized by the data aggregation in one specific corner of a given feature inter-corelation graph, while barely nothing is visible on some others. The origin of the corner is described as the first malware gene of the family and the source of sub-families. Data smoothing, where inferences about data value is presented in Fig. 6, 7 and 8. On the abscissa, densities are represented while on the ordinate, specific data values are labeled. The is filled of categorical features used is the present research dataset. The objective behind this graph is to measure every data value weight within the overall dataset scope with 12 curves(1 for each feature) . The highest probability density (0.4) is achieved in the range [0-20], with a zero accumulative rate. From up to 20, density is ruggedly decreasing. Rationally, this specific boundary([0-20]) is most likely to contain malware traffic than the benign one. Hit-her-to around the value 30000 unities, it is important to mention that traffic is still existing despite its low density.

4.1.2. Model training and performance metrics evaluations

To evaluate the performance of SVM training process on the tailored malware CIC dataset, confusion matrix, classification accuracy, error, precision,

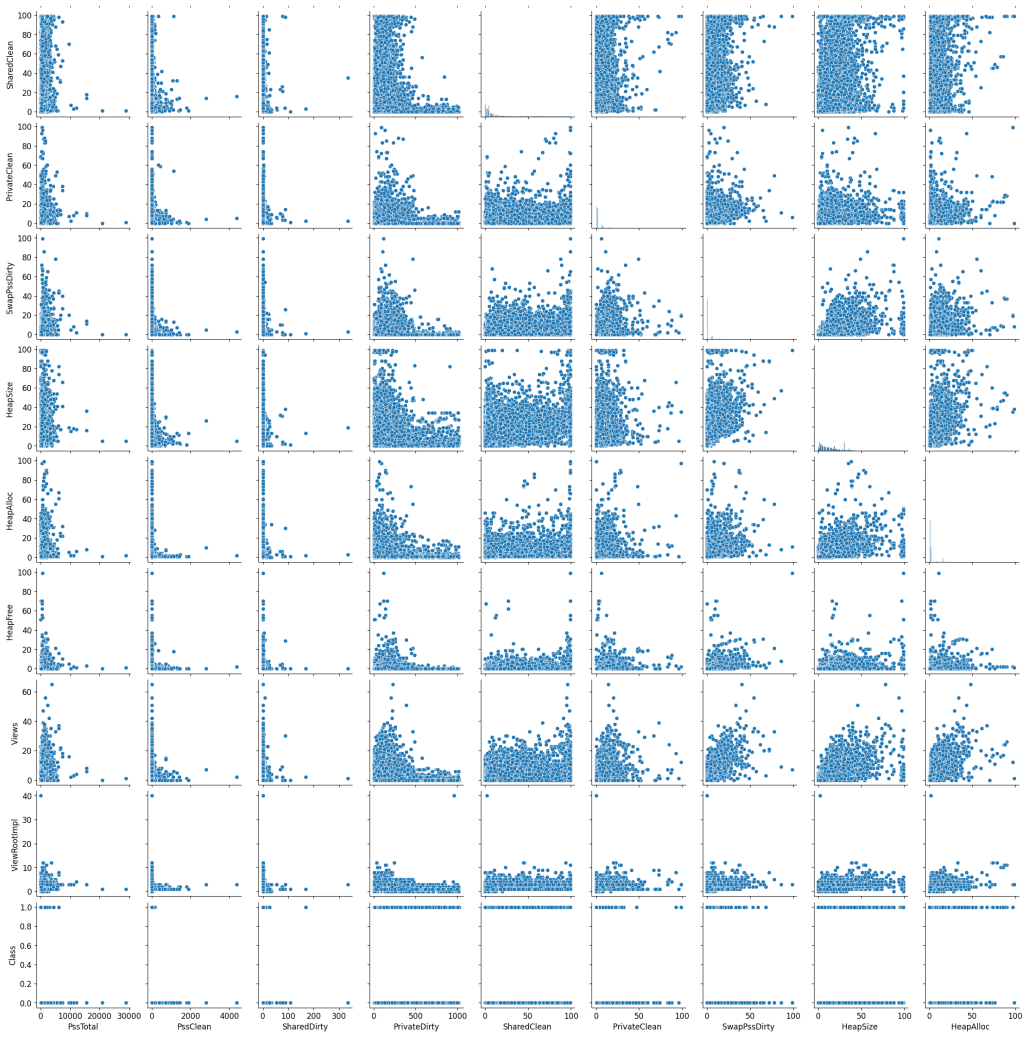

Figure 5: Scatter plot visualization

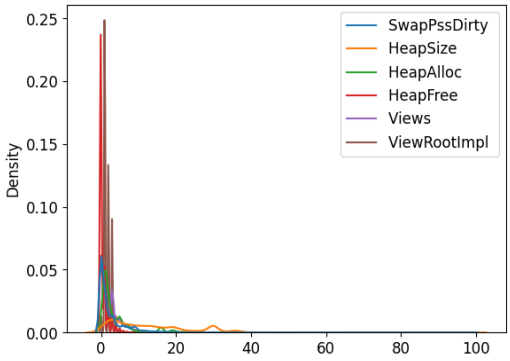

Figure 6: KDE(1)

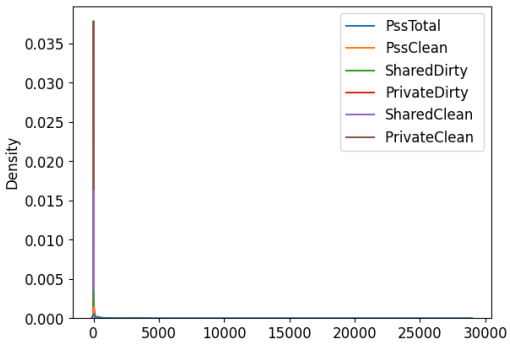

Figure 7: KDE(2)

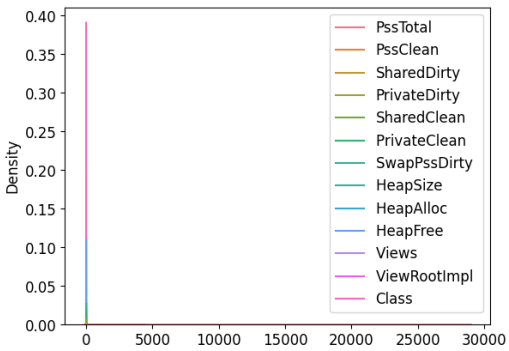

Figure 8: KDE(2)

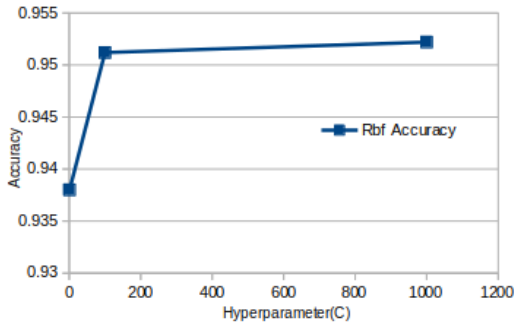

Figure 9: Ratio Basic Function accuracy score variation

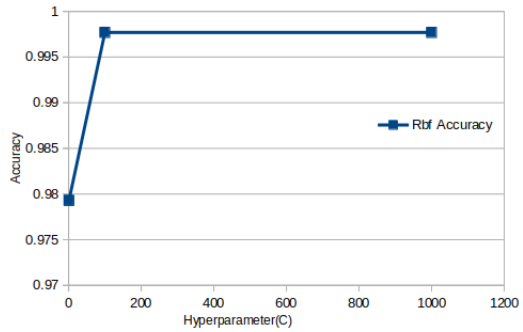

Figure 10: Objective 1

recall, f1-score, Roc-Auc score are retrieved. For the training process, respectively at hypermeter C=1,100 and 1000 unities, RBF accuracy of about 0.9380, 0.9512 and 0.9522 is presented. Fig. 9 shows the general variation of the trained model accuracy based on C's values. For C>1000, accuracy is stabilized at 0.952. By using dataset like Big Microsoft malware's dataset, results are much more interesting as presented in Fig. 10 (0.9793, 0.9977, 0.9977). Fig. 11 presents the confusion matrix of the trained model. The first big square shows four other squares representing the values of True Positive, False Positive, True Negative and False Negative. The vertical bar at the right is presenting the color variation from yellow to marine blue depending on the matrix value(1000-7000). From 11873 unique values, 7800(0.9516 of TPR or recall) True positives and 396(0.0467 of FPR or classification error) False positives malware's values. Based on this information, the F-1 score is about 0.9648. Another result, confirming and affirming the veracity of previous ones, is the Auc-Roc score, which is about 0.94, really far from

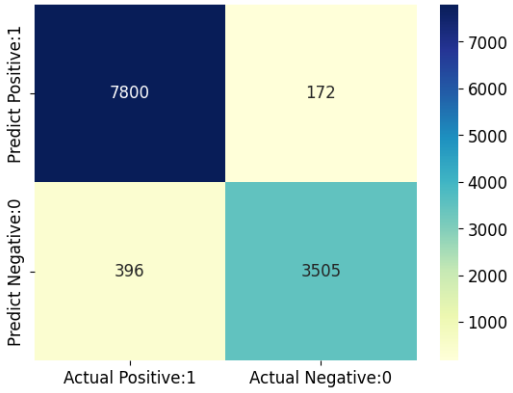

Figure 11: Confusion matrix

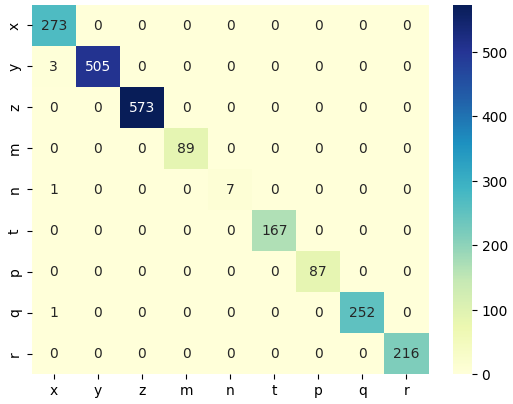

Figure 12: Objective 2

0.5. Fig. 13 presents a graphical representation of this value. The same operation was done with the MMB 2015 malware dataset as shown in Fig. 12 and 14, where results are much more convenient(TPR:0.998 , FPR:0.043, Auc-Roc:0.99).

4.1.3. Model training and performance metrics evaluations

At the post-detection phase, assuming that the event "1" is realized, the resulting "k" represents the TTP of the detected malware or a correlation of malware's TTPs in the case of inter-matching sensibility reached(polymorphism). Fig. 17 presents a glimpse from the resulting process of malware classification through FR of "x" at the pre-detection phase, giving a malware family

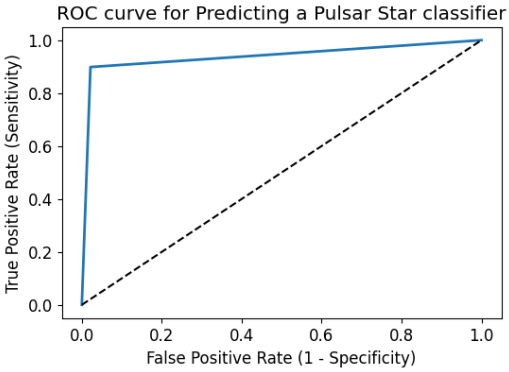

Figure 13: Roc-Auc curve

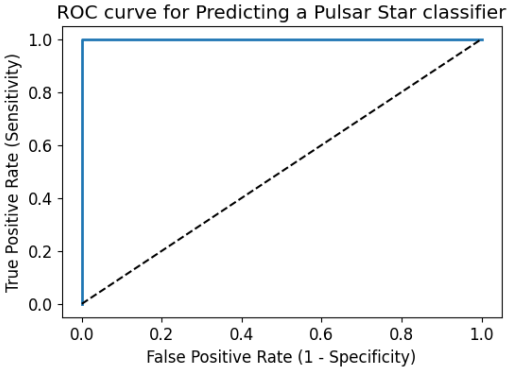

Figure 14: Objective 3

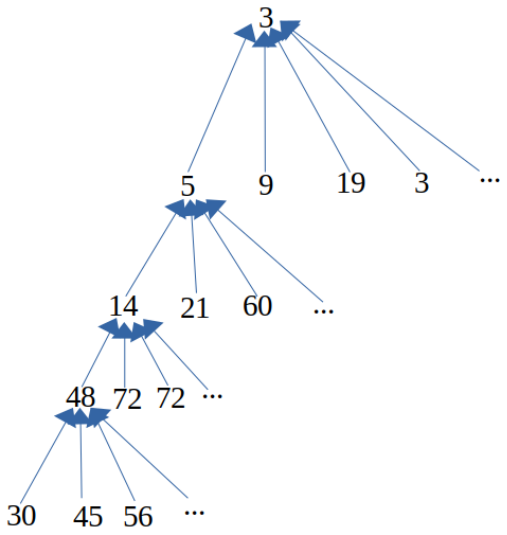

Figure 15: Index-based family graph

structure with sub-families members. Fig. 15 shows a dataset index-based representation while, Fig. 16 materialize a MD5 signature-based representation. From the collected Virus total TTPs, the ideas is to track back an originate malware parent of 2 given signatures once they are been detected within a sub-family. So if index(k1)=5 and index(k2)=9, index(k)=3, as detailed in Fig. 17. "k" shows that "k^1" and "k^2" are details about an android(executable, mobile, and apk) malware with a dangerous label of 7, called win32/ditertag.A by windows detection engine, with an infection probability of 57% and more others detail. From the beginning of this section to the end, key research result is mentioned. The next step is to discuss these present results with the existing approaches led by previous related researches.

4.2. Discussions

- **(Rq1): Affordability and the scalability of multiple ML algorithm for a probable combination**. Throughout this research activitiy pipeline, 59392 samples of malware features with 39601 malicious data and 19791 benign data, collected from the CIC Malware dataset, was trained using FR-SVM. In majority, dataset analysis operation shows a high traffic between data features based on the pulsars

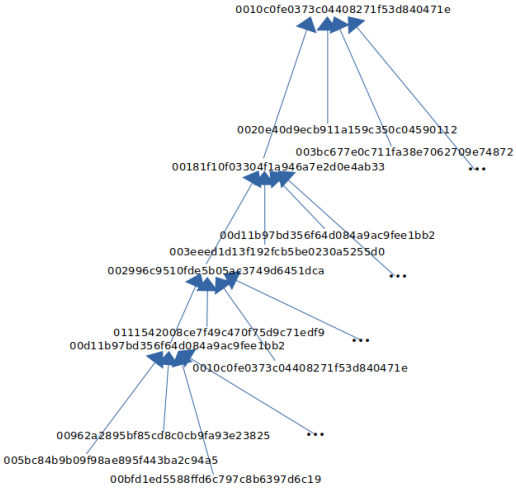

Figure 16: Signature-based family graph

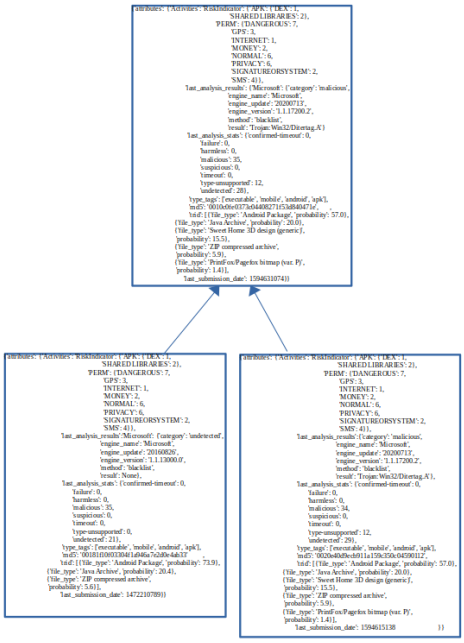

Figure 17: Value of "k" determination

plot, scatter plot and box plot. These information prove data variety of the main data source of this research scope and variability of AI mechanism that this dataset can be exposed to. Some authors did use less complex data structure[7][8][10][11][13], or much more complex [5][6][9][14] than the present one(images, and sound records includes). Another trend is the used of Generative Adversarial Networks(GAN) to auto-build a malware data structure[12] [14]. It is noticed that everything is relied on data no matter how complex is the AI algorithm(or group of algorithms) used for a given. From this point, credit can be given to HP1 assuming the data source is well provided. This hypothesis can actually work for most of the reviewed research papers presented on the related work section.

- **(Rq2): Procedure and the resources to reach an affordable precision rate on polymorphism detection**.The present research paper used the FR-SVM combination to provide a 0.952 accuracy score, 0.9516 of TPR, 0.0467 of FPR, 0.9648 of F-1 score and a 0.94 of Auc-Roc score to enhance decision on malware polymorphism. These results are better than some others findings [7][8][10][13] but not sufficient to reach what is proposed by some others[5][6][9][11][14]. The variant points are the research scope, the AI algorithms used, the dataset used, the testing environments and other prominent available research resources. By example, in [5], a DL approach for android malware 400000 malware samples dataset helps to reach 0.998 accuracy, while in [11], a Deep-Ensemble and Multifaceted Behavioral Malware Variant Detection Model using Sequential Deep Learning and Extreme Gradient Boosting Techniques was used on a dataset of about 23000 samples to reach a 0.9923 accuracy rate. In both cases, an affordable accuracy was reached but in the same way. Concerning the present research paper, an average implication on hybrid AI concept and on dataset density was taken into consideration. As the main objective was to reach affordable accuracy for malware polymorphism detection, going over stability is the next step. HP2 is true for the present case, but should be questioned in more complex challenges. Result obtained using MMB 2015 is confirming the efficiency of the proposed approach(Accuracy:0.9977, FPR:0.043, Auc-Roc:0.99). Fig. 11 presents a comparison between the proposed method and the preceding ones while Fig. 12 presents the same comparison using the Microsoft BIG Malware dataset 2015.

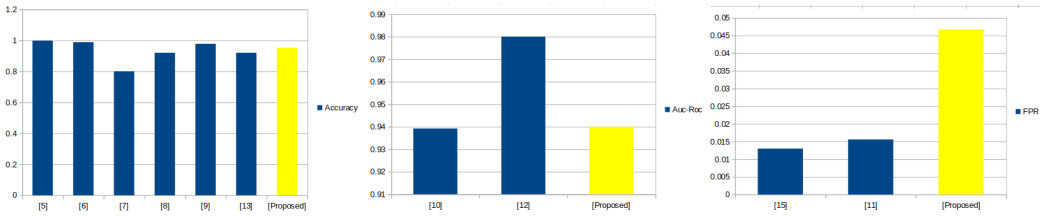

Figure 18: Subjective Research papers comparison

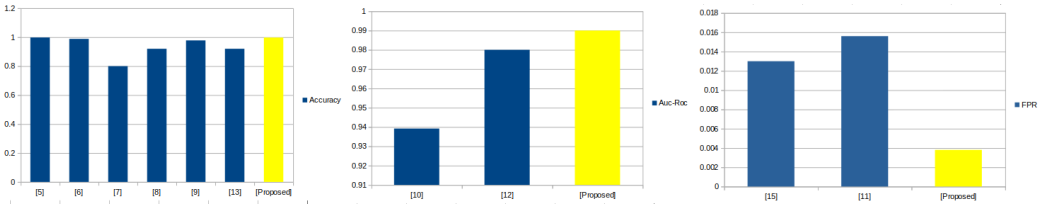

Figure 19: Objective Research papers comparison

- **(Rq3): Providing as detailed as possible real-time malware's description to enhance decision taking for their combat**. Using a fuzzy matching score of 96%, malware signature family tree was generated to emphasize the concept of malware polymorphism. Information provided by VirusTotal helped to have more description on these malware signatures. As planned in the method, precedent signatures are correlated to verify any matching polymorphism just by performing a research in the generated malware tree. The result is the value of "k" is the resulting TTP of the precedent correlation. Although Virus total was used for its signature scalability, availability and python programmability there are many other options(Mitre Attack, MISP, OSINT framework, Crowdsec, and so on). Mitre attack provides a good TTP source for cybersecurity in general but does not look over malware signature, instead it is suitable for cyber attack and group description using Structured Threat Intelligence expression(STIX). Based on this limit, Virus total or Virus share are the most indicated for the present research activity. HP3 can not receive any credit for the present research scope.

5. Conclusion

In a nutshell, the present research activity aims to tackle malware polymorphism by leveraging hybrid machine learning algorithms(RQ). As the Cybersecurity threat landscape is in continuous evolution, there is an active sense of urgency to enhance accuracy and speed in identifying potential attacks. AI can be helpful for malware's camouflage(encryption, Oligiomorphism, Polymorphic and metamorphism) and obfuscation techniques, while Threat Intelligence Platform(TIP) are essential for malware description and feeds collection. For this principal sake, CIC 2020 malware dataset was analyzed(using box plot, pulsars, scatter plot), classified(using FR at 96% of sensibility) and trained(using SVM) to achieved 0.952 accuracy score, 0.9516 of TPR, 0.0467 of FPR, 0.9648 of F-1 score and a 0.94 of Auc-Roc score. To emphasize the decision taking enhancement, polymorphism paternity of an android malware was proven by the correlation. of 2 malware signatures and their respective Virus total's TTPs to retrieve to actual father TTP. The present approach was compared with other ML/DL, Generative Adversarial Networks and advanced TIPs approaches in the wild. The proposed approach is judged as stable, efficient and reliable. It is good to notice a little divergence with the main research hypothesis(HP) in terms of the dataset used(CIC2020 instead of Malimg) due to affordability reasons. Based on actual trends and finding in the present research scope, some relevant limitations and improvements should be noted:

- 2x9 accuracy difficult to reach
- The timestamp of the overall should evaluated and improved
- The proposed method was focused at the pre-detection and post-detection phase

Compliance with Ethical Standards

Acknowledgments: The authors are very grateful to Inchtech's Team (www.inchtechs.com) for their support and assistance during the conception of that work.
Funding: We wish to confirm that there is no financial support for this work that could have influenced its outcome.
Conflict of Interest: The authors declare that they have no conflict of interest. *Ethical approval:* This article does not contain any studies with human participants and/ or animals performed by any of the authors.

References

[1] ENISA, "Malware european union agency for cybersecurity threat landscape." https://www.enisa.europa.eu/topics/cyber-threats/threat-landscape, 2024. Consulted on 20 January 2025.

[2] M. V. M. Kumar, S. L. S. Darshan, B. S. Prashanth, and V. Yarlagadda, "Introduction to the cyber-security landscape," *IGI Global*, p. 21, 2023.

[3] F. P. Appio, D. L. Torre, F. Lazzeri, H. Masri, and F. Schiavone, *Impact of Artificial Intelligence in Business and Society: Opportunities and Challenges*. 605 Third Avenue, New York, NY 10158: Routledge, 2023.

[4] T. Sarath, K. Brindha, and S. Senthilkumar, "Malware forensics analysis and detection in cyber physical systems," *IGI Global*, p. 25, 2023.

[5] Y. Wang and S. Jia, "Madras-net: A deep learning approach for detecting and classifying android malware using linknet," *Measurement: Sensors, Elsevier*, 2024.

[6] P. Maniriho, A. N. Mahmood, and M. J. M. Chowdhury, "Memaldet: A memory analysis-based malware detection framework using deep autoencoders and stacked ensemble under temporal evaluations," *Computers Security, Elsevier*, pp. 1–20, 2024.

[7] O. Olukoya and M. Fleming, "A temporal analysis and evaluation of fuzzy hashing algorithms for android malware analysis," *Forensic Science International: Digital Investigation, Elsevier*, pp. 1–22, 2024.

[8] F. M. Alanazi, B. A. Elsobky, and B. A. E. S., "A hybrid machine learning framework for security intrusion detection," *Computer Systems Science and Engineering, Tech Science Press*, pp. 835–849, 2024.

[9] T. H. Hai, V. V. Thieu, T. T. Duong, H. H. Nguyen, and E. Huh, "A proposed new endpoint detection and response with image-based malware detection system," *IEEE Access*, pp. 122859–122875, 2023.

[10] F. B. Khan, M. H. Durad, A. Khan, F. A. Khan, M. Rizwan, and A. Ali, "Design and performance analysis of an anti-malware system based on generative adversarial network framework," *IEEE Access*, pp. 27683–27708, 2024.

[11] A. A. Al-Hashmi, F. A. Ghaleb, A. Al-Marghilani, A. E. Yahya, S. A. Ebad, M. S. M. Saqib, and A. A. Darem, "Deep-ensemble and multifaceted behavioral malware variant detection model," *IEEE Access*, pp. 42762–42777, 2022.

[12] Alan and A. A. Yilmaz, "A new malware classification framework based on deep learning algorithms," *IEEE Access*, pp. 87936–87951, 2021.

[13] R. C. de Amorim and C. D. L. Ruiz, "Identifying meaningful clusters in malware data," *Elsevier*, p. 10, 2020.

[14] B. Cheng, Q. Tong, J. Wang, and W. Tian, "Malware clustering using family dependency graph," *IEEE Access*, pp. 72267–72272, 2019.

[15] J. Bai, J. Wang, and G. Zou, "A malware detection scheme based on mining format information," *Wiley, Hindawi Publishing Corporation*, pp. 1–11, 2014.

[16] S. K. Smmarwar, G. P. Gupta, and S. Kumar, "Android malware detection and identification frameworks by leveraging the machine and deep learning techniques: A comprehensive review," *Telematics and Informatics Reports, Elsevier*, p. 18, 2024.

[17] 2024. Most used Threat Intelligence platform.

[18] V. Rao, *Decision Making in the Manufacturing Environment Using Graph Theory and Fuzzy Multiple Attribute Decision Making Methods*. Springer-Verlag, 2013.

[19] W. Steeb, Y. Hardy, and R. Stoop, *The Nonlinear Workbook*. 5 Toh Tuchk Link, Singapore 596224: World Scientific Publishing, 2015.