Comprehensive Classification and Analysis of Malware Samples Using Feature Selection and Bayesian Optimized Logistic Regression for Cybersecurity Applications

Original Scientific Paper

Manisankar Sannigrahi
Vellore Institute of Technology,
School of Computer Science Engineering and Information Systems
Vellore, India
manisankar.sannigrahi2020@vitstudent.ac.in

R Thandeeswaran*
Vellore Institute of Technology,
School of Computer Science Engineering and Information Systems
Vellore, India
rthandeeswaran@vit.ac.in

*Corresponding author

Abstract *– Cyberattacks are serious threats not only to individuals but also to corporations due to their rising frequency and financial impact. Malware is the main tool of cybercriminals, and is always changing, making its detection and mitigation more complicated. To counter these threats, this work proposes a Logistic Regression approach that is based on Bayesian Optimization. By leveraging advanced techniques like a hybrid feature selection model, the study enhances malware detection and classification accuracy and efficiency. Bayesian Optimization fine-tunes the logistic regression model's hyperparameters, improving performance in identifying malware. The integration of a hybrid feature selection algorithm reduces dataset dimensionality, focusing on relevant features for more accurate classification and efficient resource use, which is suitable for real-time applications. The experimental results show amazing accuracy rates of 99.94% for the Ransomware Dataset and 99.98% on the CIC-Obfuscated Malware dataset. This proposed model performs better than the conventional detection techniques. With its flexible feature selection and optimization techniques, it can keep pace with the dynamic landscape of cyber threats. It, therefore, produces a robust and scalable answer to the current cybersecurity issues.*

Keywords*:* *Malware, Ransomware, Machine Learning, Feature Selection, Bayesian Optimization, Classification*

Received: February 14, 2025; Received in revised form: May 15, 2025; Accepted: May 15, 2025

1. INTRODUCTION

Malware today is a major threat to individuals, businesses, and governments. It refers to the viruses, worms, or other harmful programs that cause damage or exploit systems. Some of the outcomes can be very severe and may range from financial loss, data breach, identity theft, to national security threats. Another reason why the malware threat is on the rise is the sophistication of cyber attackers. Hackers develop new methods for avoiding detection and increasing the efficiency of their malware. Advanced Persistent Threats are dangerous to the critical infrastructure, health care, and education sectors. Malware requires a multipronged approach to become a threat. Organizations should be investing in holistic security approaches that include updating their software regularly, installing robust firewalls, intrusion detection systems, and training their employees. Advanced threat intelligence, along with machine learning, can enable better malware detection and mitigation. Governments and international organizations have a crucial role in formulating cybersecurity regulations and promoting international cooperation. Public-private partnerships can support the sharing of threat intelligence, thereby strengthening collective defenses against malware. Proactive and collaborative cybersecurity measures are essential to reduce risks and safeguard digital infrastructure.

Ransomware is the most widespread malware that leads to significant economic and personal losses by affecting a wide range of files of numerous organizations, personal users, and medical services. It is malware that is programmed to prevent users from gaining access to their data from the devices [1]. Ransomware looks like a normal file that infects the system from vectors like botnets, macros, and email. It remains silent inside a computer and only makes itself aware to the user after completing the encryption process. According to many ventures of cybersecurity in 2019, the total sum of money paid by the victim is 11.5 billion. Every new victim has fallen to ransomware every fourteen and eleven seconds in the years 2019 and 2021 [2]. The world is highly connected through the internet, which helps to disseminate ransomware in several protocols of communication. Ransomware has enabled attackers to launch many campaigns like Ransomware as a Service, botnets for hire, etc., to earn money by carrying out illegal activities. Ransomware has become an intrinsic part of any cyber-attack by which hackers can earn large sums of money by carrying out criminal activity [3]. The victim cannot physically remove the hard disk to any other unaffected system to access the files. The attacker asks for a payment voucher as a ransom to give the access back to the victim. A few examples of locker ransomware are CTB-locker, and Winlocker [4]. Whereas the files of the victim's system are encrypted by Crypto Ransomware, making those files inaccessible unless decrypted. Removing the hard disk or trying to remove ransomware is not going to solve anything until the victim gets the decryption key. The ransom is mainly asked in Bitcoin [5], which is widely used due to anonymity, as the attacker's identity is hard to trace. Paying a ransom never guarantees that a decryption key will be given to the victim to recover data. Many methods used to detect ransomware have low detection rates. These methods also flag benign samples as malignant and thus fail to detect malicious samples that have high false positive and negative rates. Current techniques require gathering a large amount of data by monitoring the system. The disadvantage of these techniques is that they consume a significant amount of system resources [6].

This study advances malware detection through the utilization of a hybrid machine learning model based on feature selection:

- The aim here is to enhance the effectiveness and accuracy of malware detection and classification by using features such as feature selection and hybrid models. Ultimately, the hybrid machine learning method aims to enhance the capabilities of intrusion detection systems by enhancing malware categorization accuracy.
- With a number of methods including Support Vector Machine and Naïve Bayes, malware can be investigated in comprehensive manners to study the different malware categorization approaches. Therefore, the best effective method which is best in detecting precisely classifying malware cases would be found through this project.
- This paper adds to the development of more robust and efficient methods of countering cyber threats as a result of the fusion of hybrid feature selection with the assessment of multiple methods.

The paper's subsequent sections follow this structure: Section II explores the previous works in this field. Section III examines various machine learning techniques, highlighting their strengths and limitations. In Section IV, the datasets used in the study are introduced, including details about the Ransomware dataset and CIC-Obfuscated Malware datasets. Section V delves into data visualization and feature selection techniques to enhance dataset understanding. Section VI introduces the proposed algorithm aimed at improving ransomware detection interpretability. Section VII covers the experiments conducted with the Ransomware and CIC-Obfuscated Malware datasets, presenting the results comprehensively. Finally, Section VIII provides concluding remarks and suggests potential avenues for future research.

2. RELATED WORKS

This section is dedicated to the previous literature works on malware classification and analysis. Ganfure et al. authors state that [7] ransomware attacks represent a substantial risk to businesses, but current detection methods frequently prove inadequate. The RTrap framework introduces an innovative approach employing machine-learning-generated decoy files to swiftly identify and restrict ransomware. By strategically dispersing decoy files across directories, RTrap entices ransomware, while a lightweight observer monitors these files continuously. Once detected, an automated response is activated to promptly neutralize the threat. Empirical findings underscore RTrap's efficacy, as it successfully identifies ransomware with minimal data loss, underscoring its promise in effectively countering ransomware dangers. H Bakır & R Bakır, the authors state that [8] Android malware detection has received significant attention, yet feature extraction has been relatively overlooked in machine learning-based methods. Addressing this gap, the authors introduce DroidEncoder, an innovative autoencoder-based model for Android malware classification. Using three distinct autoencoder architectures, the authors extract features from a visualized dataset containing 3000 malicious and benign Android apps. Through experiments involving various machine learning algorithms, the authors approach demonstrates superior performance across multiple metrics, validated through cross-validation. S Gulmez et al. state that [9] the escalating threat of ransomware attacks necessitates advanced detection systems beyond traditional signature-based approaches. Existing methods often rely on the machine or deep learning models to analyze dynamic features like API call sequences and DLLs. However, these methods may overlook crucial information or fail to capture the sequence relationship between features. Introducing XRan, a novel ransomware detection system, which leverages Explainable Artificial Intelligence (XAI) techniques to enhance interpretability. XRan utilizes Convolutional Neural Networks (CNNs) for detection and employs XAI models such as LIME and SHAP to provide transparent explanations. Experimental results show that XRan achieves a true positive rate of up to 99.4%, surpassing state-of-the-art methods. DW Fernando & N Komninos, the authors introduce [10] FeSAD, a framework designed to enable machine learning classifiers to effectively detect evolutionary ransomware. It comprises three layers - feature selection, drift calibration, and drift decision - ensuring reliable classification of concept drift samples. FeSAD demonstrates effectiveness in detecting drifting samples and extending the classifier's lifespan. S Sivakumar et al., the authors introduce ML-MD in this study [11], a machine learning-based strategy for categorizing malware using static methods. It employs principal component analysis (PCA) to extract dataset characteristics and introduces a Modified Particle Swarm Optimization (MPSO) algorithm for enhanced malware detection. Experimental results demonstrate the superior accuracy and detection rate of the ML-based MPSO technique compared to alternative approaches on benchmark datasets. SM Florence et al., the authors introduce [12] a machine learning classification model to combat the rising threat of crypto-ransomware. It focuses on specific network traffic features, particularly UDP and ICMP, and incorporates feature selection to improve efficiency without sacrificing accuracy. The experiment employs decision trees and random forest algorithms, combined with behavioral analysis and honeypot deployment, for effective ransomware family classification.

3. MACHINE LEARNING

It is a sub-discipline of computer science that focuses on using data and algorithms to simulate the way humans learn and incrementally improve their precision. These algorithms are used to process data, learn from it, and then make decisions, and predictions, identify patterns, and cluster based on the data collected. Machine learning can be broadly classified into three types: supervised, unsupervised, and semi-supervised learning [13]. In supervised learning, target labels and classes are known in advance, which guides the learning process. However, in unsupervised learning, the target class is completely unknown. Semi-supervised learning combines features of both supervised and unsupervised methods. The hybrid algorithm proposed in this study seeks to overcome the drawbacks of previous approaches [14]. Algorithms examined in this research are as follows, along with their advantages and disadvantages.

3.1. NAÏVE BAYES

This algorithm is a probabilistic classifier based on Bayes' Theorem [15], a statistical formula that explains the connection between conditional probabilities. Naive Bayes classification is very useful because it is fast and easy to use, especially with datasets that have many features. Bayes' Theorem calculates the likelihood of an outcome based on previous occurrences in similar circumstances. This algorithm can be explained as a probabilistic classifier which is obtained from the application of Bayes Theorem.

$$P(y|x_1, \dots\dots, x_n) = \frac{P(y)P(x_1,\dots\dots,x_n)}{P(x_1,\dots\dots,x_n)} \qquad (1)$$

In equation (1), y= Class variable, & $X1\dots\dots X_n$= Dependent Vector of features.

Naïve Bayes classifiers are good in simplicity, efficiency, and scalability. They are easy to implement, so they are good for quick deployment and prototyping. In addition, they are computationally efficient, especially for large datasets with a lot of features, due to their simple probabilistic approach [16]. Additionally, they can manage big sets of data effectively. However, these classifiers have some drawbacks, like the assumption that features are independent, which isn't always true in real data and can lead to mistakes, especially with features that are closely related. They also struggle with how features are spread out, doing very badly in cases where feature connections are complicated or when the probability method used doesn't fit the data well [17].

3.2. SUPPORT VECTOR MACHINE (SVM)

SVM is an algorithm that operates by training on a particular dataset to make precise predictions and extrapolate insights to the rest of the data. It falls under the supervised learning category of machine learning and is commonly employed for tasks such as data analysis, pattern recognition, regression, and classification. The primary goal of SVM is to identify a hyperplane within an N-dimensional space that effectively separates data points into two distinct categories [18]. SVM linear kernel function is expressed as (x, x'), which has been used for analysis

SVM has some advantages, including excelling in high-dimensional spaces and being applicable in situations with a large number of features; they are quite versatile, applicable to many kinds of data, both numeric and categorical, and in various data distributions; it resists overfitting remarkably, especially in high dimensional space, due to their ability to maximize the margin of classes; in addition, SVM can handle non-linear data [19]. SVM has some limitations, it can be computationally expensive, particularly with large datasets or non-linear kernels, due to their computational complexity; sensitivity to parameter tuning is another concern, as SVMs require careful selection of hyperparameters like kernel type and regularization parameter [20], which can greatly influence performance; their lack of interpretability poses challenges, as the decision boundary produced by SVMs can be complex and difficult to interpret, hindering understanding of the underlying decision-making process.

Fig. 1 represents the margin of SVM, which is used for the classification of data points.

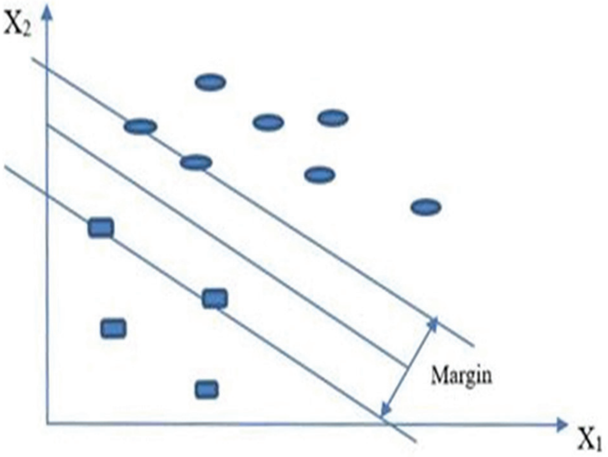

Fig. 1. Margin of SVM

3.3. RANDOM FOREST

It is a strong and adaptable tool in the field of machine learning, used for both regression and classification tasks across many applications. This algorithm creates a group of decision trees, called a "forest." Each tree in this forest is trained separately using a method called bagging. Bagging [21], in simple terms, means using the unique strengths of different models to make the group better overall. By combining the predictions of many trees, Random Forest can be more accurate and reliable than just one decision tree.

The formula of entropy is presented in equation (2).

$$Entropy = -\sum_{i=1}^{n} p_i \; log(p_i) \tag{2}$$

Information Gain = E (Parent) – E (Parent | Child), E= Entropy, p= probability.

For final evaluation, majority/hard voting method is used, the formula of this method is shown in equation (3).

$$\hat{y} = mode\ \{C_1(x),\ C_2(x), \dots \dots \dots, C_m(x)\} \tag{3}$$

Where $\hat{y}$ = class label, C_m = set of classifiers, the class label of each classifier is predicted by majority voting.

Random Forest is strong in different dimensions and typically gives high accuracy across multiple datasets, thus avoiding the overfitting phenomenon and comprehending complex patterns of the data. The strength against overfitting comes from methodologies like bootstrap sampling and random selection of features [22], which causes heterogeneity among the trees and increases generalization. But it has some limitations. High computational complexity, especially with large datasets, causes longer training time and resource usage [23].

3.4. LOGISTIC REGRESSION

This method is used for binary classification, meaning it predicts the likelihood of a yes or no result based on one or more factors. It's widely used in areas like healthcare, finance, and marketing because it's straightforward to grasp. This method uses a special function to show the relationship between the outcome and the factors, and it limits the predictions to a range from 0 to 1, which represents probabilities [24].

Imagine a dataset with pairs (x, y). Here, x is a matrix with m rows and n columns, where each row represents a sample and each column is an attribute of that sample. The y part is a list with m items, each matching a label for the samples in x. Equation (4) defines the weight matrix, which is used for generating a random initialization.

$$a = w_0 \ + \ w\grave{}_1 x_1 + w\grave{}_1 x_2 + \dots w\grave{}_1 x_n \tag{4}$$

Then pass the output to the link function which is shown in equation (5)

$$\hat{y}_i \ = \ {}^{1}/_{(1 + e^{-a})} \tag{5}$$

Then the cost function is calculated by utilizing equation (6)

$$\mathrm{cost(w)} = \left(-\frac{1}{\mathrm{m}}\right) \sum_{\mathrm{i=1}}^{\mathrm{i=m}} \mathrm{y_i} \log(\mathrm{y_i}) + (1 - \mathrm{y_i}) \log\,(1 - \hat{\mathrm{y}}_\mathrm{i}) \tag{6}$$

The updating of weights is done as per the derivative of the cost, the formulas are shown in the equation (7) and (8).

$$dw_j = \sum_{i=1}^{i=n} (\hat{y} - \hat{y}_i) x^i{}_j \tag{7}$$

$$w_{\,i} = w_j - (a\ dw_j) \tag{8}$$

Logistic regression helps us figure out the chance that a data point belongs to either class '0' or '1', using some values w and x. The key part is the exponential function inside the sigmoid function [25], which makes sure the probability is always positive. To keep the probability below one, we divide the top number by a bigger number. Equations (9) and (10) show us how to calculate these probabilities, which we then use to find the sigmoid function.

$$P = e^{w_0 + w_1\, x_1 + w_2 x_2} \tag{9}$$

$$P = \frac{e^{w_0 + w_1\, x_1 + w_2 x_2}}{e^{w_0 + w_1\, x_1 + w_2 x_2 + 1}} \tag{10}$$

Equation (10) is divided by Equation (9) to obtain the numerator term, resulting in the sigmoid function. Equation (11) defines this sigmoid function.

$$\mathrm{P} = \frac{1}{1 + \mathrm{e}^{-(\mathrm{w_1 x_1 + w_2 x_2 + \cdots + w_n x_n})}} \tag{11}$$

Logistic Regression is highly valued for its simplicity, thus being a first choice in rapid prototyping and result interpretation. Its coefficients are expressed as odds ratios, and hence, provide direct information about the effects of the predictor variables on the outcomes, which enhances interpretability [26]. In addition, it is very robust to noise and remains stable in real scenarios, making it an excellent candidate for many applications. It does have some limitations, however.

It can only capture complex variable relationships, especially in scenarios with interactions or non-linear effects [27]. Overfitting is a concern, particularly with numerous predictor variables relative to observations.

Fig. 2 represents the curve of logistic regression.

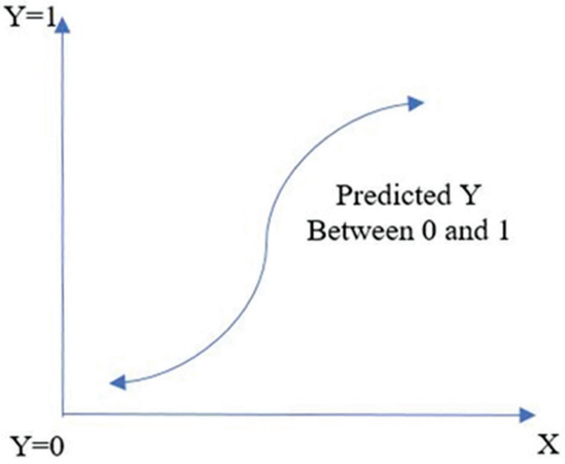

Fig. 2. Logistic Regression Curve

3.5. Logistic Regression

Bayesian optimization is an intelligent way of searching for the best parameters of complex problems without explicit formulas. It relies on a method based on probability, quickly searching through all options and finding the best for making a task better. It works very well if each option is to be tested at great cost or with significant time consumption. Unlike the grid and random search, Bayesian optimization learns the past tests to make this search faster. The main part of this method is that it starts with a probabilistic model, which creates a guess about the best settings, and then it keeps updating this guess through a process called the acquisition function [28] as it learns more. It is one of the most powerful ways to optimize functions [29], Equation (12) is used to determine the next sampling point.

$$X_t = \mathrm{argmax}_x u(X|D_{1:t-1}) \qquad (12)$$

Where, u = acquisition function, $D_{1:t-1}$= the total t samples.

There are mainly three types of acquisition functions: Upper Confidence Bound (UCB), Probability of Improvement (PI), and Expected Improvement (EI). The EI acts as a guiding metric during the optimization process, trying to balance exploration of new configurations with exploitation of the already identified good ones. It helps in an efficient search for optimal hyperparameters. Equation (13) defines the expected optimization process.

$$\mathrm{EI}[x^*] = \int_{f[\dot{x}]}^{\infty} (f[x^*]f[\dot{x}])\mathrm{Norm}_{f_{[x^*]}}[\mu[x^*]\sigma[x^*]df[x^*] \qquad (13)$$

Where, $\mu[x^*]$ = mean value of data point x, $\sigma[x^*]$= variance value of data point x, β= controlling parameter of the degree of exploration, $f[x^*\ ]$= normal distribution, $f[\dot{x}]$= current maxima.

Bayesian optimization does extremely well in optimizing functions, given its efficiency to strike a balance between exploration and exploitation, adaptability by dynamic updates of the probabilistic model, robustness with noisy data, and its ability to follow the pursuit of global optima. It converges to solutions quickly, adapts itself according to changes in the objective function landscape, does not have any problems in dealing with noisy objective functions, and can seek global optima via probabilistic predictions iteratively [30]. It has limitations regarding computational cost, sensitivity to initial conditions, surrogate model complexity, and suitability for smooth functions. It is computationally expensive, especially for large-scale tasks or complex models, thus limiting scalability. Sensitivity to initial conditions and surrogate model hyperparameters may impact its performance [31].

4. DATASET DESCRIPTION

The Ransomware dataset consists of 156 features with 1534 samples, among them 952 goodware and 582 ransomware samples of 11 different ransomware families [32]. The collected samples represent the most well-known variants of ransomware encountered recently. Each ransomware is clustered into a well-known ransomware family. Each ransomware sample was checked with VirusTotal results. Most of the ransomware samples belong to crypto-ransomware including Critroni, CryptoLocker, CryptoWall, etc. Fig. 3 encompasses the total count of instances from various ransomware families.

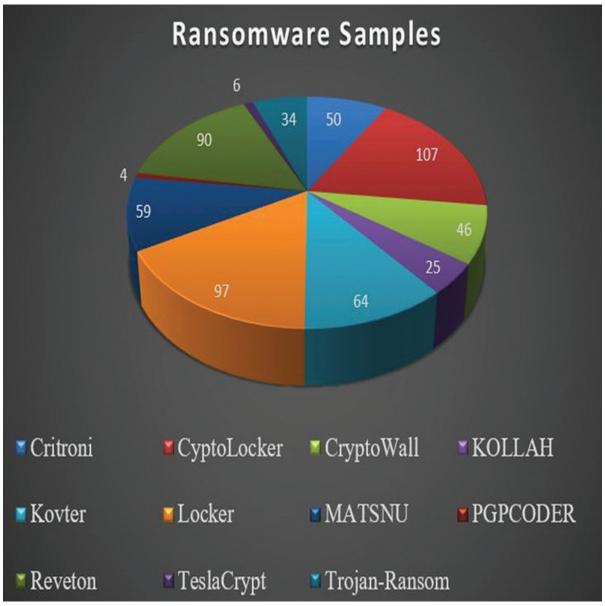

Fig. 3. Ransomware Family

CIC-Obfuscated malware dataset always focuses on representing scenarios of the real world as closely as possible by using malware that is predominant in the world. The dataset is made off of mainly three malware families Spyware, Trojan Horse, and Ransomware [33]. The dataset is being made from 50% benign and 50% malignant memory dumps. There are a total of 5832 samples with 57 features, where 2916 are malignant and 2916 are benign samples. The dataset is broken down in the Table 1.

Table 1. Description of CIC-Obfuscated Malware dataset

Malware Family	Malware Name	Count
Spyware	180Solutions	200
	Gator	200
	TIBS	141
	Coolwebsearch	200
	Transponder	241
Trojan Horse	Zeus	195
	Refroso	200
	Emotet	196
	Reconyc	157
	scar	200
Ransomware	Shade	220
	Ako	200
	Conti	200
	MAZE	195
	Pysa	171

5. FEATURE SELECTION

Feature selection is a method of aiding in the goal of creating a more accurate prediction model. This method helps in choosing features to provide better accuracy while requiring less amount of data. The main objective of feature selection is to provide cost-effective and faster predictors, improve prediction performance, and give a better comprehension of the fundamental process of generating data [34]. There are mainly three methods that are used in this paper.

5.1. VARIANCE THRESHOLD

The most simple baseline method of feature selection is the Variance Threshold. It removes the features whose threshold does not meet up and removes all features with zero variance by default. Equation (14) is utilized to calculate the variance.

$$Var[X] = p(1-p) \tag{14}$$

5.2. PEARSON CORRELATION COEFFICIENT

The measurement of the strength of the relationship between two variables and the association between them is defined as the Pearson correlation coefficient [35]. Pearson correlation is used to evaluate the linear dependency of the dataset, which is either positive or negative. The value it returns lies between -1 to 1. Equation (15) is the formula of Pearson correlation.

$$\boldsymbol{r} = \frac{\sum(\boldsymbol{x_i}-\dot{\mathrm{X}})(\boldsymbol{x_i}-\dot{\mathrm{X}})}{\sqrt{\sum(\boldsymbol{x_i}-\dot{\mathrm{X}})^2 \sum(\boldsymbol{y_i}-\hat{\mathrm{Y}})^2}} \tag{15}$$

Where, *r*= Pearson correlation coefficient, *x*= values in the *x* set, *y*= values in the *y* set, *n*= total number of values of samples *Y*.

6. PROPOSED ALGORITHM

Ransomware or malware families create major security risks to critical infrastructures. Malicious attacks cause catastrophic harm to web or mobile applications and data centers of various businesses and industries. Traditional methods are not adequate to handle sophisticated attacks [36]. In this paper, the proposed algorithm is based on Bayesian optimization and Logistic Regression algorithm. The best parameters for prediction are selected by the Bayesian optimization technique. The classification is done by optimized logistic regression. Bayesian optimization improves the performance of Logistic Regression in hybrid models by effectively tuning its hyperparameters, leading to enhanced performance and generalization. Logistic Regression relies on hyperparameters like regularization parameters and penalties, which significantly impact its functionality. Bayesian optimization efficiently navigates through the hyperparameter space to identify the best combination that maximizes performance metrics such as accuracy or F1-score [37]. Through iterative assessment of different configurations using a validation set, it steers the search towards hyperparameter values that enhance generalization. Unlike conventional grid or random search methods, Bayesian optimization dynamically selects promising configurations, resulting in quicker convergence towards optimal solutions. This adaptability proves advantageous, particularly in scenarios involving high-dimensional spaces or intricate models like Logistic Regression. Moreover [38], Bayesian optimization seamlessly integrates with ensemble techniques, further boosting overall predictive accuracy. Fine-tuning individual models within the ensemble elevates the hybrid model's effectiveness across.

Logistic Regression models require the careful tuning of multiple hyperparameters to achieve optimal performance. One of the critical hyperparameters is the Regularization Strength parameter (C). This parameter regulates the trade-off between fitting the training data

closely and preventing overfitting by controlling the strength of regularization. A lower value of C increases the regularization strength, which helps in reducing the complexity of the model and preventing overfitting, whereas a higher value of C reduces the regularization effect, allowing the model to fit the training data more closely. Another important hyperparameter is Maximum Iterations. This parameter specifies the maximum number of iterations allowed for the solver to converge. It ensures that the optimization process terminates within a reasonable time frame without compromising convergence accuracy. If the number of iterations is set too low, the solver may not converge, leading to suboptimal solutions. Conversely, setting it too high might result in unnecessarily long training times without significant gains in accuracy. Therefore, finding a balanced value for Maximum Iterations is crucial for efficient and effective model training. Random State is another vital hyperparameter. It establishes the random seed for reproducibility, ensuring consistent results across different model runs by initializing the random number generator. This consistency is particularly useful for debugging, testing, and comparing models under the same conditions. By setting the Random State, researchers and practitioners can ensure that their experiments are repeatable and that the results are not influenced by random fluctuations. Each parameter plays a significant role in balancing the trade-off between model complexity and accuracy, ensuring timely convergence, and maintaining reproducibility [39]. Proper tuning of these hyperparameters can significantly enhance the performance of Logistic Regression models, making them more reliable and effective for various applications.

Fig. 4 illustrates the framework of the proposed model. Initially, the dataset undergoes a feature selection process, after which the refined dataset is processed by the proposed model to achieve optimal classification results.

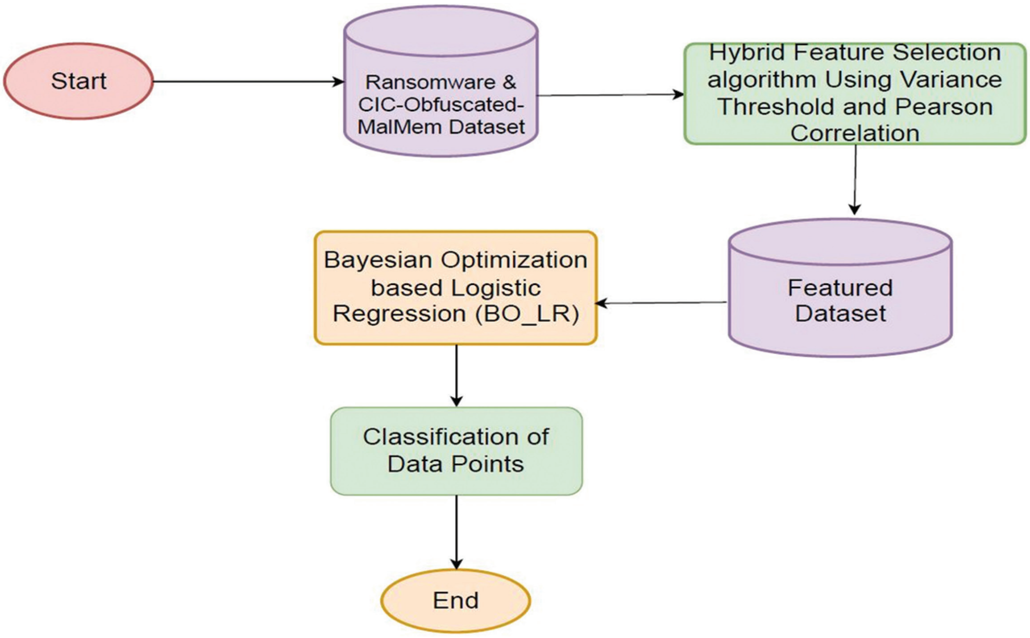
Start
Ransomware & CIC-Obfuscated-MalMem Dataset
Hybrid Feature Selection algorithm Using Variance Threshold and Pearson Correlation
Featured Dataset
Bayesian Optimization based Logistic Regression (BO_LR)
Classification of Data Points
End

Fig. 4. Workflow of Proposed algorithms over Dataset

Algorithm 1: The Proposed Bayesian-based Logistic Regression (BO_LR) Algorithm

```
Input: The dataset be $X=[X_1, X_2 \ldots\ldots X_n]$
The Target variables $Y= [Y_1, Y_2 \ldots\ Y_m]$
Output: Classification report for each target variable.
Initialize the dataset $X=[X_1, X_2 \ldots\ldots X_n]$,
    Target variables $Y= [Y_1, Y_2 \ldots\ Y_m]$, iteration $i$
Compute objective function by using Bayesian
    optimization
    $X_t$=argma$x_x\ u(X|D_{1:t-1})$
Compute acquisition function
    to select best parameters
    $EI[x^*] = \int_{f[\dot{x}]}^{\infty} (f[x^*] - f[\dot{x}]) \mathrm{Norm}_{f_{[x^*]}} [\mu[x^*], \sigma[x^*] df[x^*]$
Set $i$=0
while $i < n$ do
    Compute weight matrix, link function
        $a = w_0 + w_{`1}\ x_1 + w_{`1}\ x_2 + \ldots w_{`1}\ x_n$
        $\hat{y}_i = 1/(1+e^{-a})$
    Compute cost function by utilizing link function
        $\mathrm{Cost}(w) = \left(-\frac{1}{m}\right) \sum_{i=1}^{i=m} y_i \log(y_i) + (1 - y_i) \log (1 - \hat{y}_i$
    Update the weight
        $dw_j = \sum_{i=1}^{i=n} (\hat{y} - \hat{y}_i) x^i{}_j$
        $w_i = w_j - (a\, dw_j)$
```

```
Set $i=i+1$
end while
Calculate the Probability using sigmoid function
    $P = \frac{1}{1+e^{-(w_1x_1+w_2x_2+\cdots+w_nx_n)}}$
Return classification report for each target variable.
```

Bayesian optimization-based logistic regression provides a flexible solution to address the limitations of existing models like Naive Bayes, SVM, Random Forest, and traditional logistic regression. While traditional logistic regression struggles with non-linear patterns due to its linear assumption [40], Bayesian optimization empowers logistic regression to integrate non-linear transformations and feature engineering, thereby enhancing its ability to capture complex relationships and enhance predictive accuracy. Furthermore, Bayesian-based logistic regression tackles challenges related to noisy or irrelevant features, commonly encountered by Naive Bayes classifiers and traditional logistic regression models, through the incorporation of uncertainty estimates and robust regularization techniques. It also effectively handles class imbalances in datasets, a common issue for SVMs and Random Forests [41], by dynamically adjusting class weights or integrating sampling techniques. Crucially, Bayesian-based logistic regression maintains the interpretability of traditional logistic regression, offering stakeholders insights into prediction factors. In essence, Bayesian-based logistic regression provides adaptive hyperparameter tuning, improved non-linearity modeling, resilience to noisy data, better management of imbalanced datasets, and interpretability, rendering it a versatile and efficient approach for classification tasks [42].

The Hybrid Feature Selection (HFS) algorithm leverages the strengths of both Variance Threshold and Pearson Correlation to balance dimensionality reduction and feature diversity. This complementary strategy creates a more efficient, interpretable, and robust feature set. By integrating these two methods, the HFS algorithm enhances the performance of machine learning models, leading to improved accuracy, stability, and computational efficiency.

Algorithm 2: Hybrid Feature Selection algorithm Using Variance Threshold and Pearson Correlation (HFS)

```
Input: The dataset be $X=[X_1, X_2\ldots\ldots X_n]$
The Target variables $Y= [Y_1, Y_2\ldots\ Y_m]$
Output: Total number of column with high
        correlation value.
Initialize the dataset $X=[X_1, X_2\ldots\ldots X_n]$,
    Target variables $Y= [Y_1, Y_2\ldots\ Y_m]$, iteration $i, j$
Set variance threshold
    sel= VarianceThreshold(threshold=(.8 * (1 - .8)))
Compute variance threshold
    sel.fit_transform(X)
    sel.get_support()
    c_constant= [column for column in X.columns
    if column not in X.columns[sel.get_support()]]
Define correlation function
    def correlation(data, threshold)
Get all the names of correlated columns in a set
    col_corr = set()
    corr_matrix = X.corr()
for($i=0$, $i<$ corr_matrix.columns, $i$++)
for($j=0$, $j<i$, $j$++)
    if abs(corr_matrix.iloc[i, j]) > threshold
        colname = corr_matrix.columns[i]
        col_corr.add(colname)
    end if
    end for
end for
Compute the correlation function
        corr_features = correlation(X, 0.7)
        fea_list= list (corr_features)
selected_features= c_constant + fea_list
Return columns with high correlation and less
    threshold value
```

This step eliminates features with low variance, retaining only those that meet the variance threshold. Features with variance above the specified threshold are included in the set **c_constant**. The number of constant features removed is displayed. The function identifies features that exhibit high correlation with one another. Features with a coefficient exceeding the specified threshold are deemed highly correlated and included in the set **col_corr**. The final selection of features consists of those that passed the variance threshold and are highly correlated which are represented as **selected_features**. These **selected_features** are then returned and displayed. By following the algorithm, redundant features are effectively removed, resulting in a more efficient and interpretable dataset for subsequent analysis. By following this algorithm, redundant features are effectively removed, resulting in a more efficient and interpretable dataset for subsequent analysis.

7. EXPERIMENTAL RESULT

The suggested algorithm's detection performance is tested using datasets of ransomware and CIC-Obfuscated malware [43]. The proportion of testing samples to training samples is 70:30. The environment of the Jupyter Notebook is used for the implementation. Machine learning primarily uses two kinds of classification techniques: binary and multiclass classification. Binary classification is the process of classifying data into two groups, each designated as either zero or one.

The Ransomware dataset an initial 156 features, has been analyzed using the algorithm developed in the course of the study in order to improve the detection rate of ransomware attacks. The relationships between the attributes and the target variable can be very well ascertained from the heatmap of Fig. 5, which shows the most useful attributes for an analysis. Such an approach to the problem minimizes the number of dimensions that need to be considered, defines target variables, and enables effective detection and efficient classification models to be built.

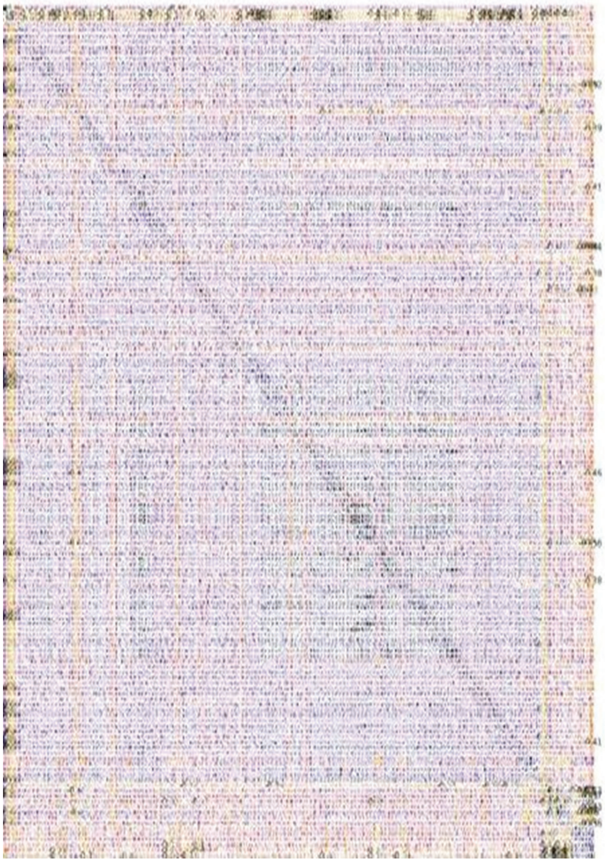

Fig. 5. Heatmap of the dataset with 156 features

The experimental results of the proposed BO_LR algorithm, compared with various traditional algorithms on the Ransomware dataset, are shown in Table 2. This comparison highlights the performance differences and demonstrates the advantages of the BO_LR algorithm over conventional methods in terms of efficiency and accuracy on this specific dataset.

Table 2. Classification of RANSOMWARE Dataset (With 156 features)

Algorithms	Accuracy	Precision	Recall	F1-Score
Logistic Regression	81%	80%	76%	79%
SVM	62%	54%	79%	64%
Random Forest	90%	91%	87%	86%
Naïve Bayes	80%	67%	80.2%	83%
BO_LR	93%	92%	92.8%	93.1%

The Fig. 6 below offers a comprehensive comparison of various evaluation criteria between the proposed model and other well-established machine learning models. It clearly illustrates how the proposed model either outperforms or matches traditional models across key evaluation metrics. By showcasing these metrics side-by-side, the figure effectively highlights the robustness, efficiency, and reliability of the proposed model compared to other machine learning approaches.

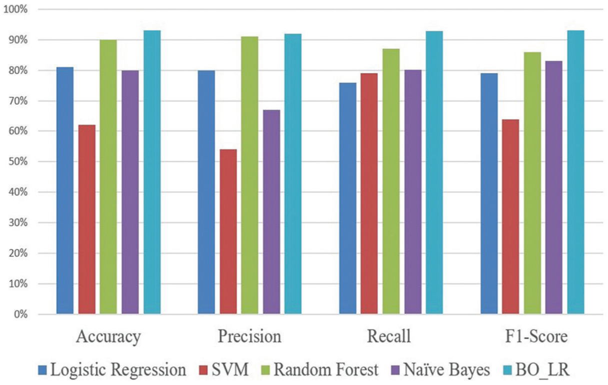

Fig. 6. Comparison over Ransomware Dataset with 156 features

By employing the proposed Algorithm 2, the hybrid feature selection algorithm, the number of features was successfully reduced to 56. This refined feature set includes those with high variance and low correlation, resulting in better outcomes compared to the initial feature set. The heatmap shown in Fig. 7 depicts the correlation between features after applying the proposed algorithm and removing unnecessary ones. This visual representation demonstrates the algorithm's success in retaining only the most relevant and non-redundant features, thereby improving the dataset's efficiency and interpretability.

The experimental results on the Ransomware dataset, which include an analysis of 56 features, are presented in Table 3. This detailed comparison showcases the performance of different algorithms on this dataset, highlighting the effectiveness of the feature selection process and its impact on the overall results.

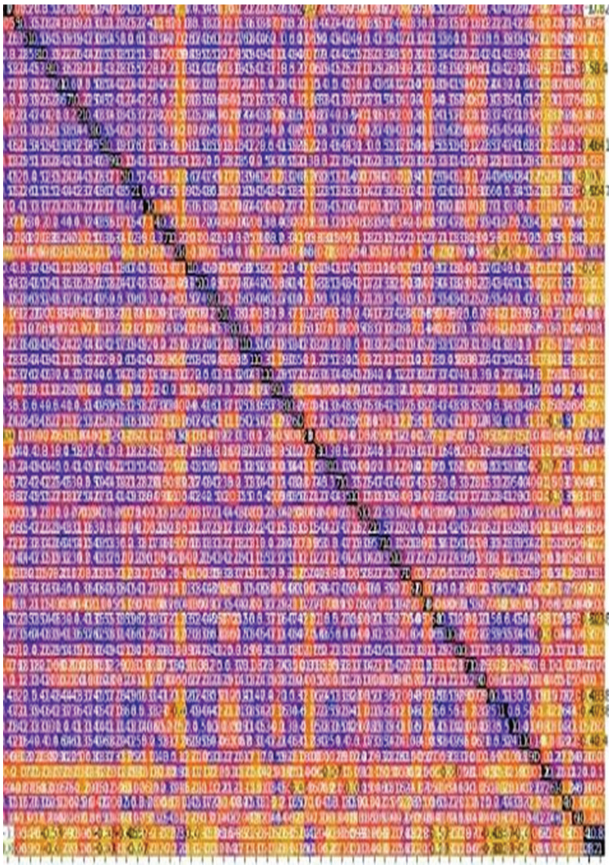

Fig. 7. Heatmap of the dataset with 56 features

Table 3. Classification of RANSOMWARE Dataset (With 56 features)

Algorithms	Accuracy	Precision	Recall	F1-Score
Logistic Regression	89%	87%	88.12%	88.3%
SVM	94%	95%	94.4%	96%
Random Forest	98%	97%	97.2%	97.4%
Naïve Bayes	91%	92%	94%	94.3%
BO_LR	99.94%	100%	99.75%	99.85%

The Fig. 8 below presents a thorough comparison of various evaluation criteria between the proposed model and other established models. It demonstrates how the proposed model either surpasses traditional models across key evaluation metrics. This detailed evaluation underscores the practical benefits of adopting the proposed model for applications requiring high accuracy and efficient real-time performance.

The second experiment was conducted on the CIC malware dataset, which comprises 57 features. The results are shown in Table 4, which were obtained using the raw dataset without applying any feature selection methods. This provides a baseline for evaluating the impact of feature selection on model performance in subsequent experiments.

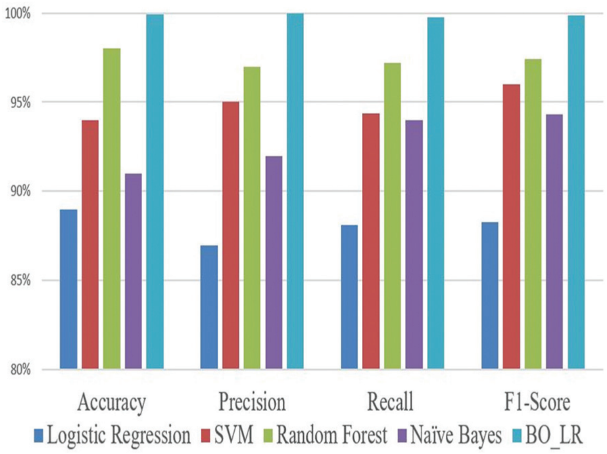

Fig. 8. Comparison over Ransomware Dataset with 56 features

Table 4. Classification of CIC-Obfuscated Malware dataset (57 Features)

Algorithms	Accuracy	Precision	Recall	F1-Score
Logistic Regression	96%	97%	95%	95.3%
SVM	94%	94.5%	92%	93%
Random Forest	97%	96%	95%	95.7%
Naïve Bayes	91%	92%	93%	94%
BO_LR	98%	99%	97%	98%

The Fig. 9 below provides an in-depth analysis of different evaluation metrics for the proposed model compared to established models using the CIC Malware dataset. It highlights how the proposed model either exceeds or matches the performance of traditional models. This comprehensive comparison emphasizes the significant advantages of the proposed model for scenarios that demand high accuracy and efficient real-time processing.

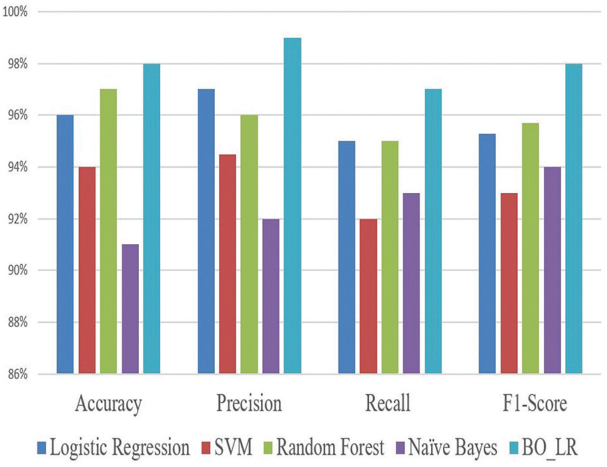

Fig. 9. Comparison over CIC Malware Dataset (57 features)

After implementing the hybrid feature selection algorithm, the number of selected features was reduced to 19. This optimized feature set preserves the most informative and significant attributes while minimizing redundancy. Table 5 below shows the performance results of the proposed BO_LR algorithm alongside traditional algorithms, evaluated on this refined feature set. By concentrating on these 19 features, the models achieve more efficient and accurate predictions. This comparison underscores the effectiveness of the hybrid feature selection in enhancing the dataset's quality, which in turn leads to superior performance of the BO_LR algorithm compared to traditional methods.

Table 5. Classification of CIC-Obfuscated Malware dataset (19 Features)

Algorithms	Accuracy	Precision	Recall	F1-Score
Logistic Regression	98%	97%	98%	98%
SVM	96%	97%	94%	95%
Random Forest	99%	98%	94%	96%
Naïve Bayes	96%	95%	93%	94%
BO_LR	99.98%	100%	99.95%	99.96%

The Fig. 10 below presents a detailed comparison of various evaluation criteria between the proposed model and other traditional models. This discussion highlights the differences in performance metrics, demonstrating how the proposed model outperforms or matches traditional models across key evaluation parameters, thus validating its effectiveness and robustness in handling the dataset.

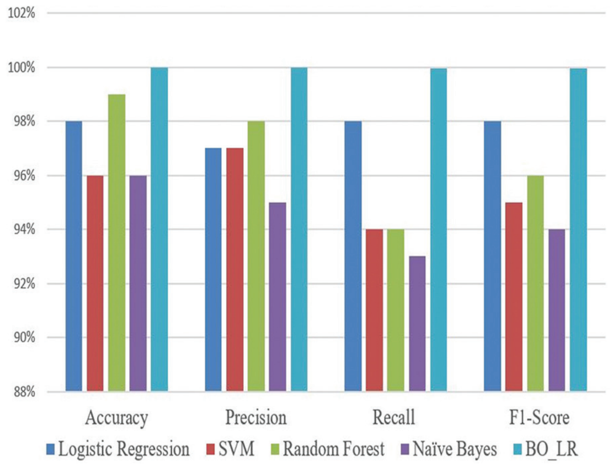

Fig. 10. Comparison over CIC-Obfuscated Malware dataset (19 features)

The data presented in Table 6 illustrates that the proposed method significantly surpasses the performance of existing approaches, confirming its superior efficiency over traditional techniques. This validation not only highlights the effectiveness of the proposed approach in achieving superior results but also emphasizes its ability to exceed benchmarks established by prior research. The findings underscore the method's innovative nature and its capacity to address the challenges associated with the dataset more effectively than existing solutions. The results bolster the case for adopting the proposed method, showcasing its potential to drive advancements in the field by offering enhanced solutions and improved performance across relevant applications. This comparative advantage suggests that the proposed method could lead to substantial improvements in practical implementations and contribute significantly to advancing current methodologies in the domain.

Table 6. Comparison over CIC-Obfuscated Malware dataset

Study	Accuracy	Precision	Recall	F1-Score
[22]	76.8%	77.3%	76.9%	76.7%
[24]	99.8%	99.5%	99.7%	99.8%
[28]	99.4%	99.7%	99.6%	99.5%
[30]	99.4%	99.43%	98.5%	98.9%
[34]	99.43%	99.17%	99.43%	99.6%
BO_LR	99.98%	100%	99.95%	99.96%

The following Fig. 11 provides a comprehensive comparison of different evaluation criteria between the proposed model and existing literature. This analysis showcases the variations in performance metrics, illustrating how the proposed model either exceeds or aligns with traditional models across essential evaluation parameters. This comparison serves to validate the effectiveness and reliability of the proposed model in managing the dataset, emphasizing its capability to deliver superior or comparable results compared to established approaches.

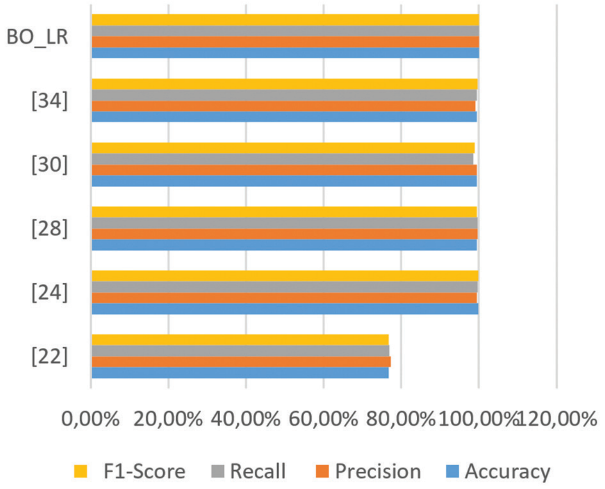

Fig. 11. Comparison over CIC-Obfuscated Malware dataset with existing literature

Bayesian optimization-based logistic regression models employ various techniques to decrease computational expenses and improve real-time applicability. They strategically explore hyperparameter space, focusing on promising regions, thereby achieving comparable or superior performance with fewer iterations, leading to reduced computational costs. Bayesian optimization identifies hyperparameters that simplify logistic regression models without compromising predictive accuracy, rendering real-time applications more viable [44]. Additionally, leveraging specialized hardware such as GPUs or TPUs accelerates the optimization process, facilitating real-time deployment [45].

8. CONCLUSION

The paper suggests a framework to identify malware through the integration of multiple machine learning methods to counter malicious threats. The framework consists of preprocessing datasets through feature selection techniques and subsequent training of machine learning classifiers to test these selected datasets. Experimentation results highlight the efficiency advantage of the Bayesian optimization-based Logistic Regression algorithm compared to other methods in detecting malware instances. The data set utilized is relatively limited and perhaps doesn't fully represent the entire set of malware variants, impacting the model's stability. In addition, while Bayesian optimization optimizes performance, computational overhead can make it inappropriate for real-time deployment in resource-constrained settings. Also missing is the implementation of deep learning, leaving the framework without validation against higher-order architectures. Besides that, the work prescribes forthcoming directions in the development of the framework, namely enlargement of the data with additional examples of malware and addition of advanced machine learning methodologies such as CNNs or RNNs. All the improvements are aimed at improving the quality and accuracy of the detection model. Lastly, the present study is intended to provide assistance in the fight against malware and improve cybersecurity defenses to be more reliable by enhancing the detection mechanism and adding advanced machine learning methods.

For future research, improving the effectiveness of the framework against zero-day malware attacks is essential. This may be done by integrating behavior-based analysis and anomaly detection techniques that enable the model to detect previously unknown threats by learning patterns characteristic of malicious behavior, instead of depending on known signatures. Increasing the dataset size to a more extensive and varied set of malware samples, including obfuscation and polymorphism varieties, would help the model generalize and be more robust. Furthermore, running the detection system in a cloud or distributed platform could greatly make it scalable and resilient to scale large amounts of data in real-time across various endpoints. Such a distributed method would also enable cooperative threat intelligence sharing holistic and future-proof solution to the continuing battle against malware.

9. REFERENCES:

[1] M. Hassan, K. Hamid, R. A. Saeed, H. Alhumyani, A. Alenizi, "Reconfigurable Intelligent Surfaces in 6G mMIMO NOMA Networks: A Comprehensive Analysis", International Journal of Electrical and Computer Engineering Systems, Vol. 16, No. 2, 2025, pp. 87-97.

[2] U. Zahoora, M. Rajarajan, Z. Pan, A. Khan, "Zero-day ransomware attack detection using deep contractive autoencoder and voting based ensemble classifier", Applied Intelligence, Vol. 52, No. 12, 2022, pp. 13941-13960.

[3] M. Alam, S. Bhattacharya, S. Dutta, S. Sinha, D. Mukhopadhyay, A. Chattopadhyay, "RATAFIA: Ransomware analysis using time and frequency informed autoencoders", Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust, McLean, VA, USA, 5-10 May 2019, pp. 218-227.

[4] S. Othmen, W. Mansouri, R. Khdhir, "Applying Artificial Intelligence Techniques For Resource Management in the Internet of Things (IoT)", International Journal of Electrical and Computer Engineering Systems, Vol. 16, No. 2, 2024, pp. 183-194.

[5] S. Poudyal, K. P. Subedi, D. Dasgupta, "A framework for analyzing ransomware using machine learning", Proceedings of the IEEE Symposium Series on Computational Intelligence, Bangalore, India, 18-21 November 2018, pp. 1692-1699.

[6] B. A. S. Al-Rimy, M. A. Maarof, S. Z. M. Shaid, "Ransomware threat success factors, taxonomy, and countermeasures: A survey and research directions", Computers & Security, Vol. 74, 2018, pp. 144-166.

[7] G. O. Ganfure, C.-F. Wu, Y.-H. Chang, W.-K. Shih, "Rtrap: Trapping and containing ransomware with machine learning", IEEE Transactions on Information Forensics and Security, Vol. 18, 2023, pp. 1433-1448.

[8] H. Bakır, R. Bakır, "DroidEncoder: Malware detection using auto-encoder based feature extractor

and machine learning algorithms", Computers and Electrical Engineering, Vol. 110, 2023, p. 108804.

[9] S. Gulmez, A. G. Kakisim, I. Sogukpinar, "XRan: Explainable deep learning-based ransomware detection using dynamic analysis", Computers & Security, Vol. 139, 2024, p. 103703.

[10] D. W. Fernando, N. Komninos, "FeSAD ransomware detection framework with machine learning using adaption to concept drift", Computers & Security, Vol. 137, 2024, p. 103629.

[11] S. Sivakumar, S. Saminathan, R. Ranjana, M. Mohan, P. K. Pareek, "Malware Detection Using The Machine Learning Based Modified Partial Swarm Optimization Approach", Proceedings of the International Conference on Applied Intelligence and Sustainable Computing, Dharwad, India, 16-17 June 2023, pp. 1-5.

[12] S. M. Florence, A. Raghava, M. J. Y. Krishna, S. Sinha, K. Pasagada, T. Kharol, "Enhancing Crypto Ransomware Detection through Network Analysis and Machine Learning", Innovative Machine Learning Applications for Cryptography, pp. 212-230, IGI Global, 2024.

[13] M. Masum, Md J. H. Faruk, H. Shahriar, K. Qian, D. Lo, M. I. Adnan, "Ransomware classification and detection with machine learning algorithms", Proceedings of the IEEE 12th Annual Computing and Communication Workshop and Conference, Las Vegas, NV, USA, 26-29 January 2022, pp. 0316-0322.

[14] W. Luo, "Network Security Situation Prediction Technology Based on Fusion of Knowledge Graph", International Journal of Advanced Computer Science & Applications, Vol. 15, No. 4, 2024, p. 881.

[15] N. Elsayed, S. Abd Elaleem, M. Marie, "Improving Prediction Accuracy using Random Forest Algorithm", International Journal of Advanced Computer Science & Applications, Vol. 15, No. 4, 2024, pp. 436-441.

[16] D. Sgandurra, L. Muñoz-González, R. Mohsen, E. C. Lupu, "Automated dynamic analysis of ransomware: Benefits, limitations and use for detection", arXiv:1609.03020, 2016, p.1609.

[17] S. K. Shaukat, V. J. Ribeiro, "RansomWall: A layered defense system against cryptographic ransomware attacks using machine learning", Proceedings of the 10th International Conference on Communication Systems & Networks, Bengaluru, India, 3-7 January 2018, pp. 356-363.

[18] S. R. Davies, R. Macfarlane, W. J. Buchanan, "Differential area analysis for ransomware attack detection within mixed file datasets", Computers & Security, Vol. 108, 2021, p. 102377.

[19] M. Hirano, R. Kobayashi, "Machine learning based ransomware detection using storage access patterns obtained from live-forensic hypervisor", Proceedings of the Sixth International Conference on Internet of Things: Systems, Management and Security, Granada, Spain, 22-25 October 2019, pp. 1-6.

[20] T. R. Reshmi, "Information security breaches due to ransomware attacks-a systematic literature review", International Journal of Information Management Data Insights, Vol. 1, No. 2, 2021, p. 100013.

[21] M. Masum, Md J. H. Faruk, H. Shahriar, K. Qian, D. Lo, M. I. Adnan, "Ransomware classification and detection with machine learning algorithms", Proceedings of the IEEE 12th Annual Computing and Communication Workshop and Conference, Las Vegas, NV, USA, 26-29 January 2022, pp. 0316-0322.

[22] D. Cevallos-Salas, F. Grijalva, J. Estrada-Jiménez, D. Benítez, R. Andrade, "Obfuscated Privacy Malware Classifiers based on Memory Dumping Analysis", IEEE Access, Vol. 12, 2024, pp. 17481-17498.

[23] A. M. Maigida, Shafi'I. M. Abdulhamid, M. Olalere, J. K. Alhassan, H. Chiroma, E. G. Dada, "Systematic literature review and metadata analysis of ransomware attacks and detection mechanisms", Journal of Reliable Intelligent Environments, Vol. 5, 2019, pp. 67-89.

[24] K. S. Roy, T. Ahmed, P. B. Udas, Md E. Karim, S. Majumdar, "MalHyStack: a hybrid stacked ensemble learning framework with feature engineering schemes for obfuscated malware analysis", Intelligent Systems with Applications, Vol. 20, 2023, p. 200283.

[25] V. Patil, P. Thakkar, C. Shah, T. Bhat, S. P. Godse, "Detection and prevention of phishing websites using machine learning approach", Proceedings of the Fourth International Conference on Computing Communication Control and Automation, Pune, India, 16-18 August 2018, pp. 1-5.

[26] A. Yeboah-Ofori, C. Boachie, "Malware attack predictive analytics in a cyber supply chain context using machine learning", Proceedings of the International Conference on Cyber Security and Internet of Things, Accra, Ghana, 29-31 May 2019, pp. 66-73.

[27] E. G. Dada, J. S. Bassi, Y. J. Hurcha, A. H. Alkali, "Performance evaluation of machine learning algorithms for detection and prevention of malware attacks", IOSR Journal of Computer Engineering 21, No. 3, 2019, pp. 18-27.

[28] M. M. Abualhaj, S. N. Al-Khatib, "Using decision tree classifier to detect Trojan Horse based on memory data", TELKOMNIKA (Telecommunication Computing Electronics and Control), Vol. 22, No. 2, 2024, pp. 393-400.

[29] A. Alqahtani, F. T. Sheldon, "A survey of crypto ransomware attack detection methodologies: an evolving outlook", Sensors, Vol. 22, No. 5, 2022, p. 1837.

[30] M. H. L. Louk, B. A. Tama, "Tree-based classifier ensembles for PE malware analysis: a performance revisit", Algorithms, Vol. 15, No. 9, 2022, p. 332.

[31] Alomari et al. "Malware detection using deep learning and correlation-based feature selection", Symmetry, Vol. 15, No. 1, 2023, p. 123.

[32] S. K. Smmarwar, G. P. Gupta, S. Kumar, "Android Malware Detection and Identification Frameworks by Leveraging the Machine and Deep Learning Techniques: A Comprehensive Review", Telematics and Informatics Reports, Vol. 14, 2024, p. 100130.

[33] A. Gaurav, B. B. Gupta, P. K. Panigrahi, "A comprehensive survey on machine learning approaches for malware detection in IoT-based enterprise information system", Enterprise Information Systems, Vol. 17, No. 3, 2023.

[34] M. Dener, G. Ok, A. Orman, "Malware detection using memory analysis data in big data environment", Applied Sciences, Vol. 12, No. 17, 2022, p. 8604.

[35] F. Deldar, M. Abadi, "Deep learning for zero-day malware detection and classification: A survey", ACM Computing Surveys, Vol. 56, No. 2, 2023, pp. 1-37.

[36] S. J. Kattamuri, R. K. Varma Penmatsa, S. Chakravarty, V. S. P. Madabathula, "Swarm optimization and machine learning applied to PE malware detection towards cyber threat intelligence", Electronics, Vol. 12, No. 2, 2023, p. 342.

[37] H. AlOmari, Q. M. Yaseen, M. A. Al-Betar, "A comparative analysis of machine learning algorithms for android malware detection", Procedia Computer Science Vol. 220, 2023, pp. 763-768.

[38] Bhat, Parnika, S. Behal, K. Dutta, "A system call-based android malware detection approach with homogeneous & heterogeneous ensemble machine learning", Computers & Security, Vol. 130, 2023, p. 103277.

[39] S. Gulmez, A. G. Kakisim, I. Sogukpinar, "Analysis of the dynamic features on ransomware detection using deep learning-based methods", Proceedings of the 11th International Symposium on Digital Forensics and Security, Chattanooga, TN, USA, 11-12 May 2023, pp. 1-6.

[40] A. Ali Almazroi, N. Ayub, "Deep learning hybridization for improved malware detection in smart Internet of Things", Scientific Reports, Vol. 14, No. 1, 2024, p. 7838.

[41] A. Vehabovic, H. Zanddizari, N. Ghani, F. Shaikh, E. Bou-Harb, M. S. Pour, J. CrichigNo, "Data-centric machine learning approach for early ransomware detection and attribution", Proceedings of the NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium, Miami, FL, USA, 8-12 May 2023, pp. 1-6.

[42] A. Buriro, A. B. Buriro, T. Ahmad, S. Buriro, S. Ullah, "MalwD&C: a quick and accurate machine learning-based approach for malware detection and categorization", Applied Sciences, Vol. 13, No. 4, 2023, p. 2508.

[43] F. Nawshin, R. Gad, D. Unal, A. K. Al-Ali, P. N. Suganthan, "Malware detection for mobile comput-

ing using secure and privacy-preserving machine learning approaches: A comprehensive survey", Computers and Electrical Engineering, Vol. 117, 2024, p. 109233.

[44] N. K. Gyamfi, N. Goranin, D. Ceponis, H. A. Čenys, "Automated system-level malware detection using machine learning: A comprehensive review", Applied Sciences, Vol. 13, No. 21, 2023, p. 11908.

[45] S. Usharani, P. M. Bala, M. M. J. Mary, "Dynamic analysis on crypto-ransomware by using machine learning: Gandcrab ransomware", Journal of Physics: Conference Series, Vol. 1717, No. 1, IOP Publishing, 2021, p. 012024.