

Hybrid Machine Learning Model for Android Ransomware Detection Using URL-Based Features

Meenakshi Jalandra¹, Megha Kuliha², Jasmeet Kaur³

dept. of Information Technology
SGSITS
Indore, India.

Abstract:

With the rapidly evolving cybersecurity landscape, ransomware – which primarily targets Android systems via malicious URLs – has become a serious concern. This work explores the use of supervised machine learning models for accurate and early ransomware detection. We compared the performance of Random Forest, Light GBM, SVM, Logistic Regression, and Naive Bayes in handling complex, high- dimensional data. The results obtained from these models were compared and light GBM and Random Forest showed accurate and robust results that were significantly better than other models. Using these results, we integrated them into a hybrid model. Deep feature interactions are better captured using Multi-Layer Perceptron (MLP) In the dynamic era of cybersecurity, the growing number of ransomware is emerging as a serious threat, especially on the Android platform, with URL threat emerging as a new fraud. This paper works on detecting URL ransomware threats in Android and presents a hybrid MAC layering model for the same. The results emphasize the effectiveness of URL type identification and multiple accurate intelligent identification of ransomware and dangerous URLs compared to traditional URL based detection. Future work includes scalable analysis and aggregation of real- time analysis. To improve the model performance and get better results, a hybrid model was built by taking the best outputs of both Random Forest and Light GBM models and combining them into an ensemble of Multi-Layer Perceptrons (MLP) and getting better outputs at different layers. This model was able to detect URLs well and used both static and dynamic URLs from the data set which proved helpful in reactive threat detection. Future extensions of this study may also include applying deep neural networks and deep learning on industrial and URL data to accurately identify URLs in real-time.

Keywords: Random Forest, Light GBM, Machine Learning, Hybrid Model, MLP, Android Ransomware, URL Detection, and Cybersecurity.

I. INTRODUCTION

With changing times, many new technologies are being invented, one of which is the smartphone. After the invention of the smartphone, the number of people using it is also continuously increased with time. In the 21st century, everyone uses a smartphone, and today the smartphone has become an important part of their life. However, as more people are dependent on smartphones is also increasing, and with the widespread use of Android-based devices, some of its disadvantages are also being faced. As we all know that every technology brings with it many adverse effects with its use; similarly, after the use of smartphones, there are many disadvantages too, one of which is the increase in cyber criminals. With the increase in the use of smartphones in every sector, cyber criminals have shifted their focus towards exploiting the smartphone mobile platform. One of the most serious threats in this area is Android ransomware, a class of malware that either encrypts user data (crypto ransomware) or locks the user interface (locker ransomware) and demands

payment from smartphone users for restoration. Many users who use smartphones are not yet aware enough to be aware of these cyber crimes, and they become a victim of this crime without even realizing it, and they have to bear heavy losses. These attacks often originate from malicious URLs, in which the URL is sent to the user through a message or email, and the cyber criminals become successful, so quick and accurate URL-based detection becomes a vital step in ransomware prevention. Ransomware is a kind of malicious software that restricts or stops users from using their system by locking their data or the screen of the device until a ransom is paid. More recent ransomware families, which are all grouped together under the umbrella term "crypto ransomware," encrypt certain file types on compromised computers and demand payment from victims via specific internet payment channels in order to provide a decryption key[3]. This category encompasses three types of malware on Android. Crypto-ransomware, PIN lockers, and lock-screen ransomware.

A picture that covers the entire screen prevents cybercriminals from accessing the user's machine in lock-screen ransomware. PIN lockers function similarly, except instead of locking the device, they alter the combination that unlocks it to a value that the user is unaware of by abusing a protective PIN that is part of the operating system. Lastly, when "crypto-ransomware" encrypts files, the data of the cybercriminal's mobile device is encrypted. Particularly when it comes to ransomware attacks, Android has become a prime target for online fraud. The term "ransomware" refers to malicious software that encrypts or locks data on a device and then demands payment to unlock it. Locker ransomware, which stops the user from accessing their device, and crypto ransomware, which encrypts its data, are the two primary ways Android ransomware manifests.

Given the increasing reliance on Android devices for both personal and professional needs, identifying ransomware is essential. Harmful URLs are one extremely dangerous attack vector. Cybercriminals often embed malware onto ostensibly harmless URLs that download malware payloads when viewed. The price or exchange rate of digital currencies, as well as the type of ransomware, affect ransom charges. Because ransomware operators believe that cryptocurrencies provide anonymity, they frequently request ransom payments in bitcoin. Newer versions of ransomware have also mentioned other ways to pay, such as using iTunes or Amazon gift cards. Please be aware, though, that paying the ransom does not ensure that the decryption key will be obtained or not. Traditional antivirus tools rely heavily on signature-based methods, which fail to detect new categories and evolving real-time ransomware threats. Such methods should be developed that can detect any type of dangerous URL in a user-friendly and smart way in less time. In contrast, machine learning (ML) techniques provide adaptive and scalable solutions by learning patterns of URL behaviour, structure, and statistical properties. Keeping this in mind, our study focuses on developing an intelligent system that classifies a given Android-related URL as malicious or benign using supervised machine learning algorithms. This work proposes a comparative analysis of several machine learning models, including Random Forest, Light GBM, SVM, Naive Bayes, and Logistic Regression to determine the most effective approach for real-time ransomware URL detection. In this study, our attempt is to further improve the performance of the model by designing a hybrid model by combining the features and strengths of the models that give the best results after analyzing all these models, which will increase both the accuracy and reliability of this project. To make this project system more user-friendly and practical, we have designed an interface in which users have to input a URL and the system will instantly give a classification result and tell whether to visit the URL or not. The simplicity of the front-end and the robustness of the back-end model demonstrate the feasibility of integrating machine learning in real-world ransomware detection applications.[17]

The crooks have total control over our desktop. One kind of virus that fits under this category is ransomware. The combination of the terms "ransom" and "malware" in the phrase accurately describes their actions. They are malicious software that requests money in return for functionality that has been taken, personally identifiable information that has been stolen, or information that the user has been refused access to. Data on the victim's computer was encrypted by the original ransomware in order to collect money for the software or key that would unlock the data. The methods used by this program to defraud its victims of their

money have evolved throughout time. It is accurate to say that ransomware is only a form of blackmail that is frequently utilized for widespread extortion.

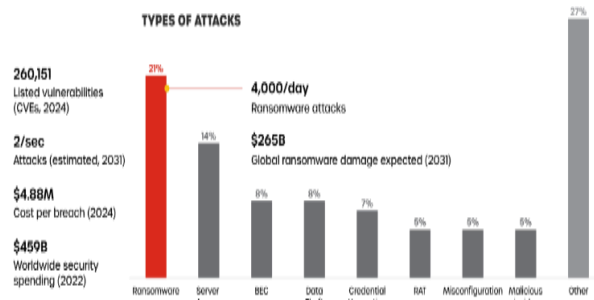


Fig.1 \$11 Trillion Yearly Losses from Cybercrime (2025) from Axiado

Ransomware Attacks have skyrocketed because of the COVID-19 epidemic, which has made people increasingly dependent on computers and internet commerce in Work from Home. ransomware assaults have skyrocketed. The notorious ransomware assault against Colonial Pipelines in May 2021 caused significant disruptions to the key petroleum supply chain activities in 17 states, including Washington, DC. The company has no choice but to pay over USD 4.4 million. JBS, the largest meat processor in the world, was also attacked during that period. These attacks impact a far broader range of businesses in addition to the government, healthcare, and educational sectors. Hackers have taken a keen interest in this relatively new virus because of its powerful attacks and potential for quick financial gain. Ransomware's objective is to stop It stops the victim from using their own resources via locking. the operating system or encrypting certain files, such as PowerPoints, spreadsheets, and images, that seem to be significant to the victim.

II. RELATED WORK

As mobile applications grow, there is significant research in the area of ransomware detection, with each new application requiring researchers to develop antivirus and prevention tools to deal with new versions of ransomware, especially on the Android platform given their wide adoption and open-source nature. Earlier approaches relied primarily on signature-based detection, which, while efficient against known threats, failed to adapt to polymorphic or zero-day ransomware variants. This limitation has fueled interest in machine learning-based methods to detect malicious behavior through static and dynamic analysis. To tackle Ransomware, apps and antivirus have been developed that can easily or precisely detect fraud URLs. As a lightweight and efficient alternative to app-level analysis, URL-based fraud detection approaches have gained popularity. []

Blacklist-based techniques are used by programs such as Fish Tank, Google Messenger, Email, Chrome's Safe Browsing, however they are often outdated and difficult to use because they do not accurately detect new types of fraud. However, in order to provide intelligent predictions, machine learning models make use of a variety of URL characteristics, including as length, entropy, character frequency, domain structure, and statistical patterns.

We looked through a lot of these studies, some of which we have discussed in the article above, and we also examined some of them, which helped us locate reliable information and techniques Al-Rimy, Maarof, and Shaid (2018) [1] presented an early behavioural detection approach that relies on dynamic analysis instead of conventional signature-based techniques to combat Oday crypto-ransomware attacks[1] In order to identify ransomware early on, their architecture keeps an eye out for suspicious activity, such as unusual file encryption and erratic resource access. After testing, it was discovered that the method works well for detecting ransomware before serious harm is done. This study highlights the need of proactive and real-time detection, offering a more flexible way to counteract advanced ransomware attacks, especially 0-day versions that get past traditional protections. SVM shown a effective detecting a URL ransomware. [11]

A thorough analysis of ransomware risks and detection methods was presented by Kok, Abdullah, Jhanjhi, and Supramaniam (2019), [2] with an emphasis on the difficulties in detecting ransomware assaults due to

their growing complexity. The study looks at several kinds of ransomware, such as crypto- and locker-ransomware, and analyzes how attackers are using new strategies. The writers examine a number of detection strategies, stressing the benefits and drawbacks of heuristic, behaviorbased, and signature-based approaches. The report also covers new developments in detection technologies, such as anomalybased methods and machine learning, highlighting the necessity of sophisticated, adaptable systems to successfully counter ransomware attacks in a constantly shifting cyber environment. [2]

Bansal (2021) [4] offered a study of ransomware assaults, offering an overview of the evolution, kinds, and impact of ransomware on cybersecurity. This article investigates the many types of ransomware, such as crypto- and locker ransomware, and looks at how hackers compromise computers. Bansal also talked on the operational and financial fallout from ransomware attacks, highlighting how common and sophisticated these threats are becoming. [3]The evaluation also emphasizes the stateofthe-art methods for detection and mitigation, emphasizing the necessity for more sophisticated and flexible security measures to keep up with the everevolving tactics of ransomware.

Alqahtani and Sheldon (2022) [5] brought attention to the constantly changing nature of these dangers. The report highlights the growing sophistication of ransomware while reviewing a variety of detection methods, including signature- based, behavioural, and machine-learning techniques. The writers go over the drawbacks of conventional approaches, namely their incapacity to combat zero-day assaults and investigate more sophisticated strategies that use predictive analytics and real-time monitoring. Their research sheds light on new developments in ransomware detection, emphasizing the need for resilient and adaptable systems to counteract the increasing intricacy of threats posed by crypto ransomware.

In order to improve detection accuracy, Herrera-Silva and Hernández-Álvarez (2023)[6] presented a dynamic feature dataset for ransomware detection. Machine learning methods are used in this dataset. In contrast to static approaches, their work focuses on the extraction of dynamic behavioural data during ransomware execution, which enables better detection. Several machine learning models were used to evaluate the suggested dataset, demonstrating its efficacy in ransomware identification. By offering a solid dataset for the development of cutting edge machine learning-based ransomware detection methods, this research advances the field.

An early-stage detection method for Android ransomware was presented by Singh and Tripathy (2024) [7] , who emphasized the significance of detecting ransomware before data exfiltration takes place. Their strategy is to identify ransomware activity early on in the assault, so averting major data loss or harm. The suggested method makes use of machine learning techniques in order to halt ransomware before it has a chance to completely carry out its destructive payload by examining early warning signals. The report emphasizes how important it is to detect ransomware early on in order to stop it from permanently damaging Android devices. Bellizzi, Vella, Colombo, and Hernandez-Castro (2022)[8] found that timing-captured memory dumps offer a novel method of detecting covert attacks on Android devices. Their focus is mostly on targeted attacks that are designed to evade traditional defenses. The recommended method looks at memory dumps captured at critical stages of an attack to identify malicious activity that could otherwise go unnoticed. The study highlights the effectiveness of memory forensics in spotting complex threats on Android machines and offers a framework for quick response to minimize any impact from stealthy hacks.

Yamany et al.[9] investigate different techniques for ransomware detection and offer a comparison of these methods. This study examines the tools, methods, and criteria employed to recognize ransomware. An indexing method for ransomware was proposed by them, which includes search functionalities, similarity checks, sample categorization, and grouping. This new approach highlights native ransomware binaries through the use of hybrid data from the static analysis system. Our system utilizes the fixed characteristics of the ransomware to monitor and organize samples, revealing their similarities. The first goal in accomplishing this is to ascertain the absolute Jaccard index. The study concludes that the performance of the IAT function exceeds that of the Strings method.

III. METHODOLOGY

Our goal is to find ways to detect ransomware attacks on Android mobiles in a low cost, which has been largely ignored so far. To reach this goal, we will use AI algorithms and methods that are aware of their surrounding environment. We will learn from the work already done and move forward to see what we can do better and more accurately.

Supervised machine learning algorithms include support vector machines (SVM), which are widely used. As per Stamp (2017), an SVM tries to identify a separating hyperplane between two labeled data classes. An SVM can use the “kernel trick” to map input data into a higher dimensional space, where the extra dimensions provide more opportunities for discovering a separating hyperplane. Each feature used in the training process is assigned a clear weight by a linear SVM. These weights indicate how important the SVM considers each feature to be (Guyon & Elisseeff, 2003). Next, we outline the feature ranking process we use, which relies on the weights derived from linear SVMs. This is the essential feature of SVMs that renders them useful for the malware. Among the research methods employed, initially we gathered 276 real ransomware samples, which provided the ammunition to test the proposed detection scheme. The idea was to engage Support Vector Machine (SVM) for learning the class of calls to APIs in ransomware as features along with which it could be used to recognize new ransomware. This solution suggested delving more profound into the application programming interface (API) call history than did subsisting solutions.

We selected this dataset because we had to select a dataset with a diversified range of URL types that we could utilize to train our models. This dataset, which we downloaded from Kaggle, has about 6728848 rows and 60 columns of differing categories (string, hostname, portname, hamming, url label source url_has_login url_has_client url_has_server url_has_admin url_has_ip,) among many others. There were some null and empty values in the unsupervised dataset. Because of the size of the dataset, it was essential to clean and render the data useful so that testing results could be generated. It included cleaning the data first and then preprocessing the data to make adjustments. This made a lot of null and useless clean datasets customized to our requirements, and this dataset was then used in various machine learning models. Nonetheless, we rejected some of the columns that we trained since they did not need a large number of variables.[19] We are defining them here:

```
(1) y_test = test_data['label']
X_test = test_data.drop(columns=['url', 'source', 'url_entropy', 'url_hamming_1', 'url_hamming_0',
                                'url_hamming_10', 'url_hamming_100', 'url_hamming_1000', 'url_entropy',
                                'path_count_no_of_calls', 'path_count_lower',
                                'path_count_upper', 'domain_min_distance', 'label'])

X_test_scaled = scaler.transform(X_test)

# Get RF predictions
rf_preds_test = rf_model.predict_proba(X_test_scaled)[:, 1].reshape(-1, 1)
```

Fig. 2 drop columns details for model

We needed to choose one that was well constructed since we did not have much time for this study and there were many models. We made a quick change and made it usable. We proceeded to preprocess this dataset, clean the data, and then train it on the Random Forest model, which gave us very outstanding accuracy. As shown in this research, our accuracy was better than (6), showing that our model is very competent in being able to predict the URL and identify whether it is safe or not. the data frame and its associated images to extract the relevant information from the dataset.

We ran each of the datasets separately on each of the models (Light GBM, Naive Bayes, logistic regression, Support Vector Machine, etc.) and the results were different for each one of them. Some models performed well based on the result analysis, and some others performed poorly due to the data set being binary. We

constructed a hybrid model which we discuss in the hybrid portion of the paper where we described the performance measures of the models which performed better. The effectiveness of the proposed scheme was demonstrated in a sandbox environment, which testified was done via malicious PE file analytics in both datasets. Preprocessing had been done in EMBER, as it provided pre processed feature vectors, but additional preprocessing was required on FFRI to select certain features. The custom loss function allows the addition of two cost sensitive weights, α and β , to reduce false positives (FP) and false negatives (FN) by assigning higher penalties during the training phase of the models. The parameters α and β were finetuned with a heuristic approach to optimize the performance metrics.

Let.

- PRF (x) = Random Forest probability prediction for input URL x
- PLGBM(x) = Light GBM's probability estimate for the same input
- H(x) = the final feature vector supplied to MLP
- FMLP(H(x)) = The final results for MLP (ransomware/safe categorization)
- The hybrid feature input to MLP is defined as Follow: $H(x) = [PRF(x), PLGBM(x)]$ (1)

Final classification output

$$Y = FMLP(H(x)) = FMLP([PRF(x), PLGBM(x)]) \dots (2)$$

Where,

$Y \in \{0,1\}$, safe URL , 1 = Ransomware URL

FMLP is Multi layer Perceptron function during training. Feature importance had calculated using RF, and low-impact columns were removed. Dataset was imbalanced, class weight were adjusted.

IV. PROPOSED HYBRID MODEL

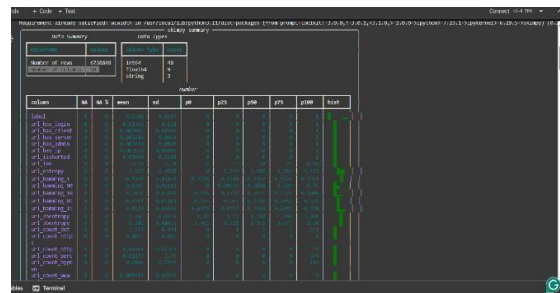


Fig.1 data frame



Fig.4 Data Frame Details

as improved by better detection of ransomware. This research is facilitated by tailored log loss function, employed effectively within the Light GBM algorithm for malware detection. Validation of the method comes from two datasets, FFRI (not public) and EMBER (public). Feature extraction

In order to increase the accuracy, robustness and flexibility of ransomware URL detection, we used several models in this research. In conclusion, we got different results from all the models. In the results obtained from these models, the accuracy of some models was found to be different. So we tried to create a hybrid model by combining their parameters, in which we got quite good success. Initially, we only trained the models and analysed their results. We got different values for all the model parameters like accuracy, precision, recall, f-1 score, support, in which we figured out that we can create a hybrid model by using the models giving high accuracy. For which we demonstrate that this study provides high accuracy of two high performance machine learning classifiers: Random Forest (RF) and Light GBM (LGBM). After seeing that both these models generate multiple decision trees from the data present in the random forest data set by randomly chunking a set of features from each tree and explaining all the possibilities available at a time and can capture non-linear patterns as well. It is robust especially on high-dimensional data and performs better on imbalanced or noisy data. And light GBM (light gradient boost machine) works well on data sets that have a large number of features and samples. Also, it worked fast and efficiently on our large data set which is its specialty. Also, dart (drop outs multiple additive regression trees) randomly ignores some trees along with every occurrence. It uses overfitting which increases our accuracy and also handles missing values on datasets like tabular data and it uses low memory as well as gives fast training.

As seen in Figure X, this ransomware detection has a complex structure of Multi-Layer Perceptron (MLP). MLP takes as its input the future probability feature vector from Random Forest and Light GBM classifiers through a hybrid model (Random Forest and Light GBM) and generates its output based on the analysis of different dynamic and static URL-based URLs. An input layer, two hidden layers, and an output layer that use a sigmoid function to determine if the URL is safe or malicious make up this structure. A ReLU

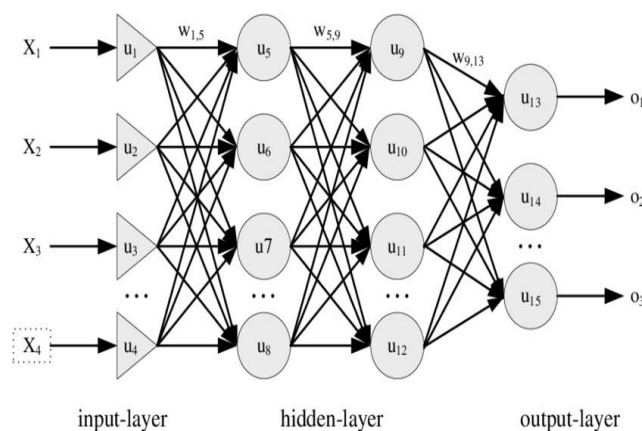


Fig. 5 Figure x The hybrid model uses a Multi-Layer Perceptron (MLP) architecture to classify ransomware URLs by integrating predictions from Random Forest and Light GBM.

We combined the predictions of all the trees we built in it. Due to this feature of both these models, we combined their parameters and used soft strategy. To make the hybrid model better, MLP (multilevel perceptron) is a neural network-based parameters and used soft strategy. To make the hybrid model better, MLP (multilevel perceptron) is a neural network-based model that does non-linear combination of features and can approach the function in a better way and tell its accuracy. The reason for integrating MLP with RF and light GBM is that we can add facilities like possibility and prediction like activation function is also used by every neuron in all of its levels. According to the figure, MLP consists of three levels: an input layer, two fully linked but concealed layers, and an output layer. Regarding the layers, only one value from each base model is accepted by the input layers. Additionally, the 64,32 neurons that make up the hidden layer employ the Rectified Linear Unit (ReLU) function to create a complicated and non-linear network that allows for linear learning.

their output to MLP. We can know the hidden interaction between random forest and light GBM which we cannot know from a single model, meaning we can easily know the intricacies of these models of random forest and light GBM through MLP which may be difficult for a single model. At the same time, we can also work with MLP as a meta-learner to see how it provides the final prediction in an accurate manner. We can learn this technique by model stacking or ensemble learning where the output of more than one model can be given as the input of a single model and we can also get the results which were not available.[18]

we came to know that hybrid-MLP model is better at detecting ransomware We can detect false positives and false negatives without giving too many false positives and false negatives Along with this, the MLP model here can also learn complex results that we get from the combination of RF and LGBM. Being a hybrid model, the accuracy of detection increases and we can also find false negatives Research also supports that ensemble + deep learning is more helpful in detecting many fraudulent URLs, malware.

- Eg. Layers : input \square 64 \square 32 \square output
- Activation ReLU (this is hidden) and Sigmoid (this is output)

Criteria	Individual RF/Light GBM	Hybrid Model
Accuracy	was high but it was not looking perfec	Hybrid got better on the data set because we compared both the models
Bias/variance	Both the different models were balanced	Here it looked good in some places
Feature learning	We gave features by doing manual manipulations	We were also able to give features of deep learning in hybrid from MLP in which we had to give individually
Generalized	Either it was underfit or overfit	Being a NN layer or ensemble was doing very well

In this study, we examined the suggested hybrid model Light GBM (prediction and robustness), Random Forest (strong feature comprehension and prediction), and a lightweight MLP layer (RF to enhance classification based on Light GBM output). Each URL receives a score from Random Forest, and Light GBM also classifies URLs as safe or unsafe. After MLP analyses the features gathered from the URLs, the next layer

Table 1

classifies them as 64, 32, and 1, providing the final prediction. A hybrid model is then run to determine whether the URL is malicious or helpful. Here, we have prioritized the goal of the model operating precisely and fixing errors. The model is displayed in a number of metrics and uses a radar map to illustrate how it operates.

V. DESIGN AND SCOPE OF THE SOLUTION

This solution proposed the proper path to detect ransomware even more focused by how hackers use different methods to access URLs on Android devices. And we can prevent devices from attacks using different secure methods and can apply models to fix problems before hackers get to them. Due to the increase of apps and mobile applications, fraud is increasing day by day. These are systematic ways to prevent users' devices. For context awareness to be implemented, a context ontology must be established and attack profiles created. The data are input into algorithms that employ artificial intelligence, allowing for a conclusion to be drawn regarding the impending attack. Contextual information such as Debug Size, Debug RVA, Major Image Version, Major OS Version, Export Size, IAT RVA, Major Linker Version, Minor Linker Version, Number of Sections, Size of Stack Reserve, DLL Characteristics, and Bitcoin Addresses is utilized to study the dataset. Four main elements comprise the solution: data gathering, a context ontology for feature extraction, attack context filters a classification algorithm for prediction making, and an alert regarding the outcome [1]. Fig. 1 illustrates the proposed model, while Fig. 2 depicts its working flow

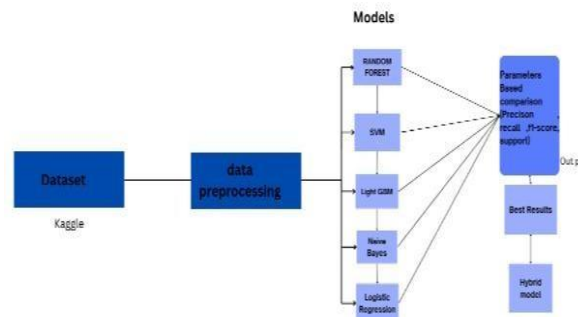
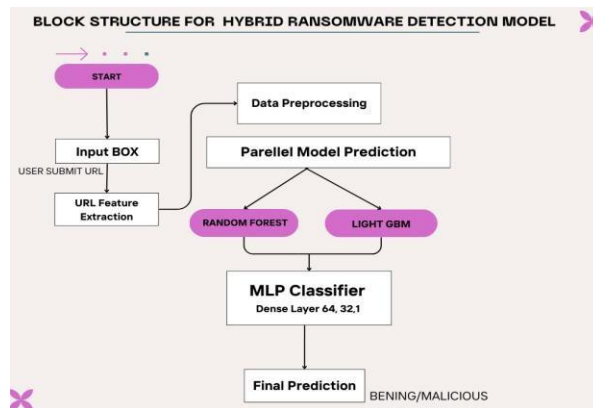


Fig. 6 models diagram

Structured Parts of the Way to Solve the Problem-

- Network stacks and packet capturing tools collect data initially. These tools observe the data transmission and reception process of the internet of things devices. Data collected is then used for training that includes routine traffic and traffic from ransomware. Upon a JSON file, the details relating to the



testbed are kept.

Fig.7 Block Structure for Hybrid Ransomware Detection Model

- Context ontology makes data gathering simple from the structure and logic of representation. Context ontology is fed all these things by the data gathering unit when found
- A classification method such as support vector machine (SVM) processes the feature vectors. It serves to detect the assault and trigger a warning. In SVM modeling, it is necessary to find the optimal hyperplane for data classification. The optimal hyperplane fully utilizes the data margin of the training set. The training set is made up of n observations of the form (x_i, y_i) , with x_i being a p -dimensional vector. The smallest value of

$\|w\|$ will correspond to the largest margin.

VI. RESULT ANALYSIS

This study delves into multiple machine learning models, such as Support Vector Machines (SVM), Random Forests (RF), Light GBM, Naive Bayes, and Logistic Regression, for ransomware detection using data obtained from Kaggle. It also uses other various sources from literature. This study examines the efficiency of each model in ransomware detection through the use and development of static and dynamic ransomware features created off of some datasets.

A. EFFECTIVENESS OF SELECTED MODEL

- Support Vector Machines (SVM): SVM is such a valuable method to carry out ransomware detection because of the manifold ability of this kind of classification in the high dimensional spaces. In fact, research by Takeuchi et al. (2018) has reached the conclusion that SVM can be used with high success also in lowering the number of FP while obtaining a significant TPR, particularly with dynamic characteristics such as API calls and system.
- Random forest- An interpretation of Khammas (2020), it's also known for the power to handle imbalanced datasets. It secures the terrific levels of detection rates through aggregation of decision trees and conclusively choosing the highest probability of classification. This representation also points out to the fact that its accuracy is still consistently high in 90s across various ransomware data sets because it performs sufficiently well in storing both static and dynamic attribute mining.
- Light GBM- Although for the huge data, Light GBM as revealed by Gao et al. (2022), is more effective in many cases. Right now, custom loss functions can be designed and incorporated-meaning such as the possibility of use of cost-sensitive log loss, which gives confidence that an improved misclassification rate means lower false positives (FP) and false negatives (FN) since it provides a greater positive outlook.
- Naive Bayes: As is seen in studies by Ramadhan et al. (2020), it remains a simple well-qualified tool for inferring the findings from the data, indeed noting down the pros and cons of predictive ability in the method. In a sense, even if it is one of the most simplistic models present in machine learning having a very basic approach to classification, naive Bayes usually drifts toward the effective and results in competitive outcomes through the use of a very carefully chosen few features and produce great results in not dramatically heavy detection systems.
- Logistic Regression: In conventional terms, Logistic samples Higher $TPR \in [0,1]$ value indicates the good performance of the machine learning model

TP

Regression models were created to manage regression tasks, whereas they contribute to binary classification. In essence, this method is the knowledge base on $TPR(Recall) = \frac{TP}{TP+FN}$

(4)

important characteristics of such capacities without fully providing the same accuracy as more advanced classifiers like Light GBM and RF in capturing False Positive Rate (FPR): The percentage of benign samples wrongly classified as ransomware:

TP

nonlinear relations and complex data performance. Precision =

$$\frac{TP}{TP + FP}$$

(Actual results) (5)

The importance of decreasing the false positives managed by Light GBM through the cost-sensitive loss function. (5) effectiveness of machine learning methods in the identification of ransomware with dynamic feature datasets, as does our result, showing an SVM and RF connection between a very high TPR. (6) forgiveness of predicting models for identifying ransomware with AI establishes that ensemble techniques, including RF and gradient boosting for classification improvement, are very effective. (7) Indeed, the data set provided is an enabling factor in terms of assessing models to comprehend balanced effects.

To judge the efficiency of chosen machine learning models— Support Vector Machines (SVM), Random Forest (RF), Light GBM, Naive Bayes, and Logistic Regression—performance metrics, namely accuracy, True Positive Rate (TPR), False Positive Rate (FPR), and Area Under the Curve (AUC) will be used to assess how well the software identifies ransomware attacks while preserving a balance between the false alarms and correct classifications

Criteria	Predicted (ransomware)	Predicted (benign)
Actual (ransomware)	TP	FN
Actual (benign)	FP	TN

Formulations of Exceptions that are Scientific

Then, the evaluation metrics are mathematically described as:

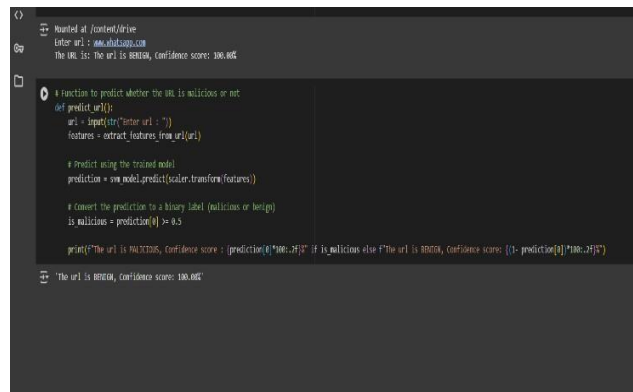
Accuracy: It refers to the proportion of the correctly classified samples.

Area Under the Curve (AUC): A score indicating the trade off between TPR and FPR across various thresholds. F1 score computes trade-off in between Precision and Recall, provide an harmonic means of precision and recall

$$\text{Precision} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (\text{Predicted results}) \quad (6)$$

Precision × Recall

Here we attached some ss of implementation and working of model:



```

def predict_url(url):
    url = input("Enter url : ")
    features = extract_features(url)

    # Predict using the trained model
    prediction = svm_model.predict(scaler.transform(features))

    # Convert the prediction to a binary label (malicious or benign)
    is_malicious = prediction[0] > 0.5

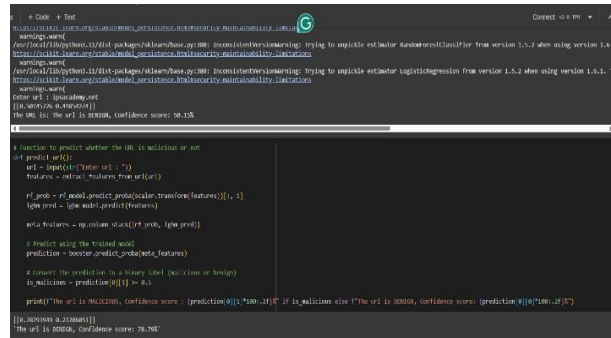
    print("The url is MALICIOUS, Confidence score : (prediction[0]*100:.2f)%" if is_malicious else "The url is BENIGN, Confidence score : (1- prediction[0])*100:.2f)%"

    "The url is BENIGN, Confidence score: 100.00%"
  
```

Fig.8 Logistic regression defines that URL is safe or unsafe

Where,

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (3)$$



```

# Reactor to predict whether the URL is malicious or not
def predict(url):
    url = urlparse(url)
    features = extract_features(url)
    rf_prob = rf_model.predict_proba(scaler.transform(features))[1, 1]
    type_prob = type_model.predict(features)
    url_features = np.concatenate((rf_prob, type_prob))
    # Predict using the trained model
    prediction = booster.predict_proba(url_features)
    # Convert the prediction to a binary label (malicious or benign)
    is_malicious = prediction[0][1] > 0.5
    print("The url is %s, confidence score : (prediction[0][1]*100-50)%" if is_malicious else "The url is %s, confidence score: (prediction[0][0]*100-50)%"
    (is_malicious, url))
    "The url is %s, confidence score: 78.78%"

```

Fig.9 SVM defines that URL is safe or unsafe

TP- The True Positives or proper identification of ransomware TN- The Truly Negatives or proper identification of benign files.

FP- The False Positives, or the benign files being incorrectly identified as ransomware.

FN- The False Negatives, or the ransomware being incorrectly identified as benign.

True Positive Rate (TPR) or Sensitivity or Recall: To find out the percentage of proper recognition of actual ransomware

The observed results from the assessment of the observed dataset here discuss about every model accuracy , performance which shows that TPR and FPR of every model which are used here for comparison based on dataset we found some values which mention here in tanle, which gives detects ransomware are summarized in the table(2) below:

Model	Accuracy	AUC	TPR	FPR
Support Vector Machines (SVM)	81.2%	0.82	73%	3%
Random Forest (RF)	95.7%	0.94	91%	2%
Light GBM	92.1%	0.82	82%	1.8%
Nave Bayes	79.3%	0.83	80%	5%
Logistic Regression	84.1%	0.88	82%	5.5%
Hybrid Model	93.5%	0.88	88%	2.2%

Table 2

A prototype interface that allow users to submit URLs was developed , the system instantly classifies the URL as either benign or Harmful after analyzing the input using the hybrid model. This interface ehnhances usability and highlights the real-time capabilities of our system. The combination of Light GBM offers the highest AUC that decreases FP rates which again tells that it's much appropriate for larger-scale ransomware detection systems. We found some graphs after processing dataset on every models which are following:

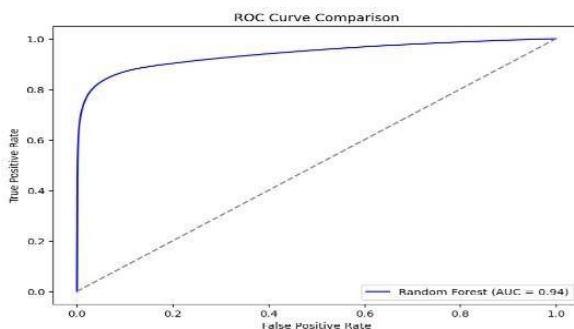


Fig.13 Naïve Bayes

Fig.10 Random Forest

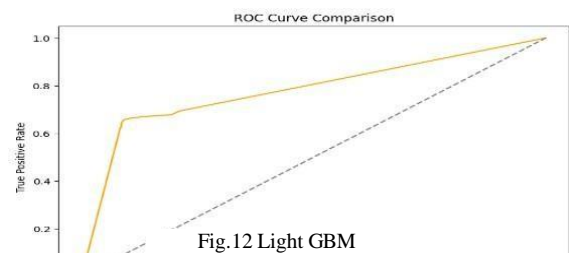
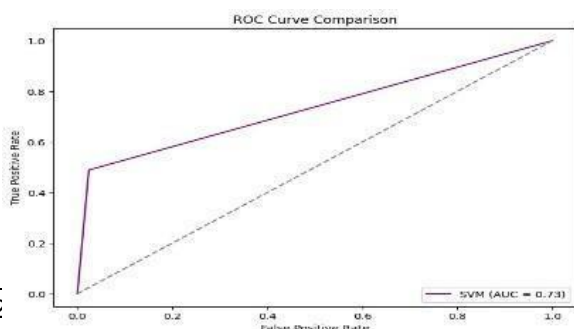


Fig.12 Light GBM

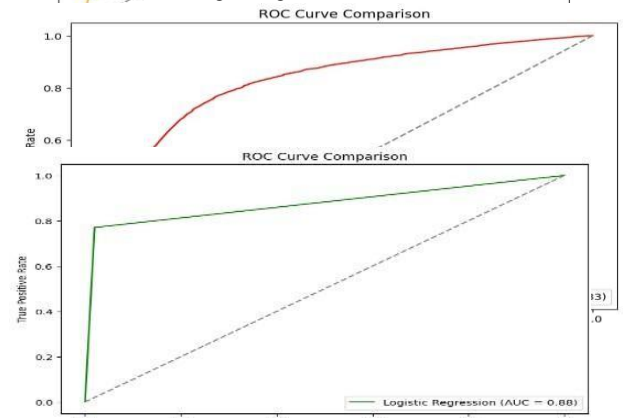


Fig.14 Logistic Regression

Fig.11 Support Vector

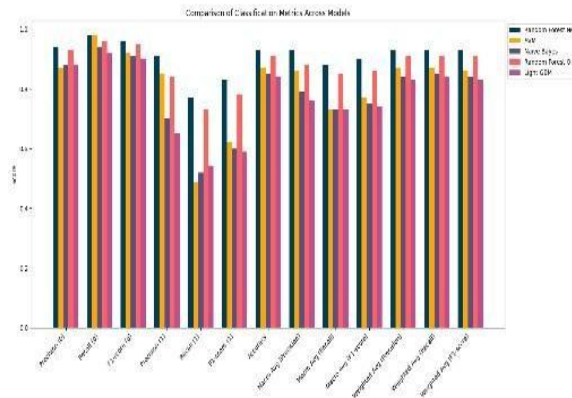


Fig.15 Combine Graph All Models

job well in such high-dimensional datasets. Besides, we will get an excellent result using Naive Bayes and Logistic Regression in some lightweight systems, but they have gained the worst underperform in heavy conditions.

This study verifies the efficiency of advanced machine learning methods in ransomware detection through the Kaggle Dataset and its rich comparison with related methodologies and how these methods are deployed in the real – world Cybersecurity system. We also plot a combine radar graph for all models.

VII. CONCLUSION

The inferences do reveal that highly accurate measures and performance in terms of model robustness are generally found in high-graded models like Light GBM and Random Forest. As Logistic regression and Naive Bayes are more recommended for systems that are lightweight and minimize requirements, the Kaggle Dataset was capable in laying down a good foundation for judging model performance on this exhaustive range of features which could provide more profound insights into machine learning efficiency. This work assesses the use of machine learning models for the identification and prevention of ransomware attacks, with an eye to their relevance in the face of the growing complexity and pervasiveness of ransomware attacks. In all the models that are herein reported, both LightGBM and RF (RandomForest) have shown the most optimal performances, not only because of their robustness to big data, It is possible to obtain highly accurate results with low false positive rate and false negative rate. In addition, the integration of costsensitive custom loss function in LightGBM contributed to improved effective errors correction between benign and malicious samples. RF most excelled in the presence of imbalanced datasets based on its ensemble of decision tree models, maintaining accuracy even in the most complex cases. the MLP model here can also learn complex results that we get from the combination of RF and LGBM. Being a hybrid model, the accuracy of detection increases and we can also find false negatives Research also supports that ensemble + deep learning is more helpful in detecting many fraudulent URLs, malware, While these advanced models showed superior performance, simpler models like Logistic Regression and Naïve Bayes also proved crucial, especially for lightweight systems where computational efficiency is preferred over accuracy. While these models suffered from some limitations in dealing with nonlinear relationships and high- dimensional data, they showed competitive results in resource-constrained environments, which makes them feasible in specific use cases. This article used data, which was sourced from Kaggle as well as from other data release repositories, as input in the study and as a tool for training and evaluation of the models. This presented static and dynamic characteristics of ransomware (e.g., API logs and system calls), allowing the models to identify the vulnerabilities during an early stage of the life cycle. The emphasis on dynamic feature extraction made it possible to detect ransomware attacks, particularly—

one of the most important criteria for damage reduction and for delay minimization as much as possible. To evaluate the performance of the models, metrics such as accuracy, TPR, FPR, and AUC were used, and RF achieved an impressive 90.7% accuracy with a low FPR of 2%.

Although LightGBM is slightly less accurate in some scenarios, its ability to produce false positives and false negatives has been clearly shown to be a flexible loss function capability by LightGBM, hence, LightGBM may offer a viable alternative for a ransomware detection system at an industrial scale. The paper reported the shift from the previous signaturebased (signature-based detection) detection methods to the recent behavioural-based (behavioural-based detection) detection method, machine learning-based (machine learning-based detection) detection method. On the one hand, the SVM-based models, for example, exploited the high-dimensional space by having a better understanding of the data and, for instance, the cost-sensitive loss of LightGBM, as a valid idea, with respect to the modeling of the complexity of ransomware behaviors. This progress sets the groundwork for the application of AI/machine learning in cybersecurity as a more flexible and billable option compared to routine approaches. Practical consequences of the results are also particularly relevant in areas such as medicine, finance, and government, in which ransomware attacks can have a severe impact. Due to the LightGBM and the RF's excellent ability of dealing big data and having high accuracy, they are finally selected to build a large scale system. In contrast, more basic models (e.g., Logistic Regression, and Naïve Bayes) would supply a good result for the small-scale formats where they continue "flat" with low capacity and consequently these can be beneficial in terms of ease of maintenance and fit for purpose. Promising (but based on the studies and evidence suggesting some limitations, including a hit-or-miss, computationally demanding, at a high level nature, or ransomware, ones unknown at the time of attack). We got accurate results from all model and using best results we made an hybrid model to use it accurate and very relevant to ransomware detecting. These in turn emphasize the need for a dynamic update of the feature sets and models, that allow the features to keep up with the increasing complexity of the attacks. In the paper, the field's role is discussed in depth as well as the aspect of taking in consideration the importance of the development of more general datasets (i.e., datasets of which a very large number of families of ransomware variants can be covered) to find, not only the generalization and robustness of the model, but also the extent to which this model could be adapted to unseen ransomware families. In future directions of studies, the deep learning based system is incorporated into the real-time tracking system and this can significantly improve the tracking performance. Analysis and predictive analytics applied to these methods can offer a more proactive approach to ransomware protection by helping organizations be more prepared for attacks. Hybridization, as an emerging area of research that combines the strengths of different types of models, is a research view that may offer an integrated protection solution. Here such framework is then readily available to leverage the robustness of LightGBM and RF, and the efficiency and ease (i.e., low complexity) of LR and NB, and being able to integrate them according to the operation requirements and constraints, respectively.

The effect of ransomware detection is not restricted to the attacked organisations, as the attacks are progressively reaching more and more critical infrastructures and public services. Critical service damage can be avoided or caught early by early detection, and thus. The damage inflicted by ransomware can be substantially reduced, so that the smooth operation of the operations can be ensured with very limited interruption. These findings support the urgency to invest in next generation cybersecurity solutions and embrace a proactive perspective toward threat detection. In conclusion, this research establishes the effectiveness of machine learning models in ransomware detection; LightGBM and RF are the top choices because of their superior accuracy, adaptability, and robustness. Even though more complex models can be used to provide reasonable substitutes in small computing environments, the integration of sophisticated algorithms and hybrid world models of network structures represents a conceivable design to bring about the new generation of ransomware security measures and defenses. The possibilities and risks of this work can be exploited by organisations for the creation of a more resilient cybersecurity infrastructure designed to prevent both the flexible approach of ransomware targeting valuable assets, and as a way to achieve a safer URL.

REFERENCES:

1. B. A. S. Al-Rimy, M. A. Maarof, and S. Z. M. Shaïd, "A 0-day aware crypto-ransomware early behavioral detection framework," in *Proc. 2nd Int. Conf. Reliable Information and Communication Technology (IRICT)*, 2017, pp. 758–766.
2. S. Kok, A. Abdullah, N. Jhanjhi, and M. Supramaniam, "Ransomware, threat and detection techniques: A review," *Int. J. Comput. Sci. Netw. Secur.*, vol. 19, no. 2, pp. 136, 2019.
3. M. Anghel and A. Racautanu, "A note on different types of ransomware attacks," *Cryptology ePrint Archive*, 2019.
4. U. Bansal, "A review on ransomware attack," in *Proc. 2nd Int. Conf. Secure Cyber Computing and Communications (ICSCCC)*, 2021, pp. 221–226.
5. A. Alqahtani and F. T. Sheldon, "A survey of crypto ransomware attack detection methodologies: an evolving outlook," *Sensors*, vol. 22, no. 5, p. 1837, 2022.
6. J. A. Herrera-Silva and M. Hernández-Álvarez, "Dynamic feature dataset for ransomware detection using machine learning algorithms," *Sensors*, vol. 23, no. 3, p. 1053, 2023.
7. N. Singh and S. Tripathy, "It's too late if exfiltrate: Early stage Android ransomware detection," *Computers & Security*, vol. 141, p. 103819, 2024.
8. J. Bellizzi, M. Vella, C. Colombo, and J. Hernandez-Castro, "Responding to targeted stealthy attacks on Android using timely- captured memory dumps," *IEEE Access*, vol. 10, pp. 35172–35218, 2022.
9. B. Yamany, M. S. Elsayed, A. D. Jurcut, N. Abdelbaki, and M. A. Azer, "A new scheme for ransomware classification and clustering using static features," *Electronics*, vol. 11, no. 20, p. 3307, 2022.
10. B. Ramadhan, Y. Purwanto, and M. F. Ruriawan, "Forensic malware identification using naive Bayes method," in *Proc. Int. Conf. Information Technology Systems and Innovation (ICITSI)*, 2020, pp. 1–7.
11. Y. Takeuchi, K. Sakai, and S. Fukumoto, "Detecting ransomware using support vector machines," in *Proc. 47th Int. Conf. Parallel Processing*, 2018, pp. 1–6.
12. Y. Gao, H. Hasegawa, Y. Yamaguchi, and H. Shimada, "Malware detection using LightGBM with a custom logistic loss function," *IEEE Access*, vol. 10, pp. 47792–47804, 2022.
13. B. M. Khammas, "Ransomware detection using random forest technique," *ICT Express*, vol. 6, no. 4, pp. 325–331, 2020.
14. S. Poudyal, K. P. Subedi, and D. Dasgupta, "A framework for analyzing ransomware using machine learning," in *Proc. IEEE Symp. Series on Computational Intelligence (SSCI)*, 2018, pp. 1692–1699.
15. D. W. Fernando, N. Komninos, and T. Chen, "A study on the evolution of ransomware detection using machine learning and deep learning techniques," *IoT*, vol. 1, no. 2, pp. 551–604, 2020.
16. S. Alsoghyer and I. Almomani, "Ransomware detection system for Android applications," *Electronics*, vol. 8, no. 8, p. 868, 2019.
17. S. Sharma, R. Kumar, and C. R. Krishna, "A survey on analysis and detection of Android ransomware," *Concurrency Computat. Pract. Exper.*, vol. 33, no. 16, p. e6272, 2021.
18. A. Ferrante, M. Malek, F. Martinelli, F. Mercaldo, and J. Milosevic, "Extinguishing ransomware—a hybrid approach to Android ransomware detection," in *Proc. Int. Symp. Foundations and Practice of Security (FPS)*, 2017, pp. 242–258.
19. I. Almomani, R. Qaddoura, M. Habib, S. Alsoghyer, A. Al Khayer,
20. Aljarah, and H. Faris, "Android ransomware detection based on a hybrid evolutionary approach in the context of highly imbalanced data," *IEEE Access*, vol. 9, pp. 57674–57691, 2021.
21. T. Soni, "Robust Android security: A random forest-based approach to malware detection," in *Proc. 4th Int. Conf. Sustainable Expert Systems (ICSES)*, 2024, pp. 1215–1218.
22. A. A. Taha and S. J. Malebary, "Hybrid classification of Android malware based on fuzzy clustering and the gradient boosting machine," *Neural Comput. Appl.*, vol. 33, no. 12, pp. 6721–6732, 2021.

23. A. A. Albin Ahmed, A. Shaahid, F. Alnasser, S. Alfaddagh, S. Binagag, and D. Alqahtani, "Android ransomware detection using supervised machine learning techniques based on traffic analysis," *Sensors*, vol. 24, no. 1, p. 189, 2023.
24. S. I. Bae, G. B. Lee, and E. G. Im, "Ransomware detection using machine learning algorithms," *Concurrency Computat. Pract. Exper.*, vol. 32, no. 18, p. e5422, 2020.
25. T. C. Miranda, P. F. Gimenez, J. F. Lalande, V. V. T. Tong, and P. Wilke, "Debiasing Android malware datasets: How can I trust your results if your dataset is biased?" *IEEE Trans. Inf. Forensics Security*, vol. 17, pp. 2182–2197, 2022.
26. X. Li, J. Liu, Y. Huo, R. Zhang, and Y. Yao, "An Android malware detection method based on AndroidManifest file," in *Proc. 4th Int. Conf. Cloud Comput. Intell. Syst. (CCIS)*, 2016, pp. 239–243.
27. V. Senthilkumar, M. Devasenan, J. Karan, C. Ravina, V. Sneka, and
28. T. Kavipriya, "Permission-based Android malware identification," in *Proc. 2nd Int. Conf. Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, 2023, pp. 1–6.
29. G. Wang and Z. Liu, "Android malware detection model based on LightGBM," in *Recent Trends in Intelligent Computing, Communication and Devices: Proc. ICCD 2018*, Springer, 2020, pp. 237–243.
30. D. T. Dehkordy and A. Rasoolzadegan, "A new machine learning- based method for Android malware detection on imbalanced dataset," *Multimedia Tools Appl.*, vol. 80, no. 16, pp. 24533–24554, 2021.
31. L. Suhuan and H. Xiaojun, "Android malware detection based on logistic regression and XGBoost," in *Proc. 10th Int. Conf. Software Engineering and Service Science (ICSSESS)*, 2019, pp. 528–532.
32. C. Irawan, T. Mantoro, and M. A. Ayu, "Malware detection and classification model using machine learning random forest approach," in *Proc. 7th Int. Conf. Computing, Engineering and Design (ICCED)*, 2021, pp. 1–5.
33. K. Shaukat, S. Luo, V. Varadharajan, I. A. Hameed, and M. Xu, "A survey on machine learning techniques for cyber security in the last decade," *IEEE Access*, vol. 8, pp. 222310–222354, 2020.
34. O. Wahyunggoro, A. E. Permanasari, and A. Chamsudin, "Utilization of neural network for disease forecasting," in *Proc. 59th ISI World Statistics Congress*, 2013, vol. 549.
35. S. Alzahrani, Y. Xiao, S. Asiri, J. Zheng, and T. Li, "A survey of ransomware detection methods," *IEEE Access*, vol. 13, pp. [to be updated if known], 2025.
36. K. Alemerien, M. Almseidin, E. Altarawneh, and M. Alksasbeh, "Autonomous defending approaches based on machine learning for Android malware detection," in *AI-Driven Security Systems and Intelligent Threat Response Using Autonomous Cyber Defense*, IGI Global Scientific Publishing, 2025, pp. 325–374.
37. H. Alamro, W. Mtouaa, S. Aljameel, A. S. Salama, M. A. Hamza, and A. Y. Othman, "Automated Android malware detection using optimal ensemble learning approach for cybersecurity," *IEEE Access*, vol. 11, pp. 72509–72517, 2023.
38. R. Fuller, C. Moore, T. Taylor, and C. Anderson, "A novel hybrid machine learning approach for real-time ransomware detection using behavior-driven heuristic features," *unpublished*, 2024.
39. M. A. Latif, Z. Mushtaq, S. Rahman, S. Arif, S. N. F. Mursal, M. Irfan, and H. Aziz, "Oversampling-enhanced feature fusion-based hybrid ViT-1DCNN model for ransomware cyber attack detection," *Comput. Model. Eng. Sci. (CMES)*, vol. 142, no. 2, pp. [to be inserted], 2025.