Detect by machine learning whether an attack using ransomware has occurred

Author: Orel Rahum
Supervisor : Dr. Amit Dvir, Harel Berger

Abstract

Ransomware is one of the most prevalent malicious software in 2020, it encodes the files in the victim's device, then demands money, i.e., ransom, for decrypting the files. The financial losses and global damage cost of individuals and organizations due to ransomware is increasing year by year. Hence, fighting against ransomware is an important issue. In this article, we proposed the ransomware detection approach based on machine learning. Our main objective was to examine with a human eye by the visible changes, we compare the encrypted text files with unencrypted files and with customized features and trainer to teach it to identify text files that have been attacked. For this purpose, we use 4 different datasets which contain thousand of text files. Each text file has 50% unencrypted text and 50% text encrypted by 6 types of encryption techniques – Atbash, Autokey, Caesar, Gronsfeld, Playfair and RSA. We trained two different machine learning classifiers – support vector machine and near neighbor algorithm – to train our model. From the experimental results, the classification accuracies of 91% were achieved with both classifiers.

Keywords: Ransomware, Machine learning, Malware detection, Security.

INTRODUCTION

Since the origin of the Internet, the world has seen its merits and the wonders. 2.5 quintillion bytes of data are generated by internet users every day. With the invention of smartphones, tablets and computers, along with innovations in mobile networks and WiFi, the creation and consumption of data are continuously growing. It is easy to see that as the size of the information that goes through the internet is greater, the need to prevent attacks over the Internet increases with it. One kind of malware that use the importance of the internet called ransomware. Ransomware has become a substantial global threat. It is a kind of malicious software that restricts users from accessing their personal and system files, in order to regain access it demands a ransom payment.

Ransomware is a comparatively new malware, due to its successful attack and direct financial interest, it has generated much interest from cybercriminals. The main objective of

Ransomware is to prevent the users from accessing their own resources by locking the operating system or encrypting targeted files that seem important to the user, such as spreadsheets, images and presentations [1]. Generally, there are two types of Ransomware: crypto and locky. The crypto ransomware uses encryption technique by locking the particular files from user access; this is much more difficult to resolve and the damage caused may be irreversible. Crypto ransomware is also the more popular type employed by cybercriminals. Whereas, locky ransomware locks the entire system from access by its user, but it is usually easy to resolve. In the literature, the third type of ransomware called scareware has been mentioned [2]. This ransomware doesn't actually harm the user's device but only scares the target into. An example of crypto ransomware is is shown in Figure 1.

Figure 1: Crypto Ransomware

Related Work

In this section, we analyse the existing work in detection of locked files. The existing approaches to detect ransomware may be ground into three main categories as follows.

1. Detection by the hardware
2. Detection by the dedicated software
3. Detection by the internet traffic

It has been noticed that, most of the solutions are based on 'use of probability ratio' according to different parameters. In the article ***2entFOX A Framework for High Survivable***,

the authors have used Bayesian network that contains common scenarios. An example of a common scenario is:

1. Access to cryptographic libraries
2. Access to specific registry paths
3. Targeted files search key words

The default setting of the software is to take 20 different options and then it calculates the probability using the Bayesian network. The probability number is between 0 to 1, if the probability is high, it means we are attacked.

In another article, ***Detecting crypto-ransomware in IoT networks based on energy***, a machine learning based approach is presented, which employed knowledge of the power consumption of each device after installing popular applications. In this article, we will build a learning machine based technique, that works on a large stock of data-set that contains benign phone data and encryption phones data. Thus, if the device went through ransomware attack, the model can predict based on the energy consumption.

In the article, ***CryptoLock (and Drop It) Stopping Ransomware Attacks on User Data***, the authors presented a model which is Cryptolock software, based on indicators (indicator is a variable that can be 0 or 1)

The software is based on 3 main indicators-

1. Change the file type - The signature define the order and location of the particular byte values unique to the type of file. Such data should be considered as suspicious.
2. Copy source file and comparison - When copying it should be similar to probability 0, if not - its suspect
3. Shannon Entropy - Use of the Shannon indicator that updates whether there is any uncertainty about the files

Cryptography

The word cryptography consists of two words. Crypto which means secret and graphics which means spelling. Cryptography is a field thousands of years old and to some extent today's cryptography is like the cryptography of yesteryear. As in any field, there are people who take advantage of the industry and there are people who make malicious use of it, which we will talk about in the article. In this article, we will focus on the field of encryption in cryptography and explain the number of types of encryption and the differences between them.

Symmetrical encryption

In the world of symmetric encryption, it is assumed that two people who want to communicate encrypted have a key that allows them to encrypt messages and decrypt or encrypt messages or verify messages. Figure 2 presents an overview of symmetrical encryption.

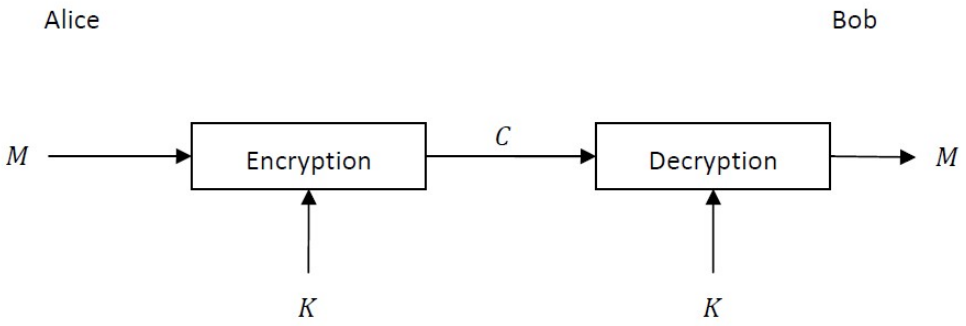


Figure 2: An overview of symmetrical encryption.

Caesar cipher

In cryptography, the Caesar cipher is also called as Caesar's cipher, Caesar's code, shift cipher, or Caesar shift. It is one of the simplest and most commonly known encryption approaches. It is a sort of substitution cipher, where each letter of the plain text is replaced by a letter with some fixed number of positions from the alphabet. For instance, using the left shift of 3, E would be replaced by B, F would become C, and so on, as shown in Figure 3. Julius Caesar used it in his private communication, that's why it was named after him.

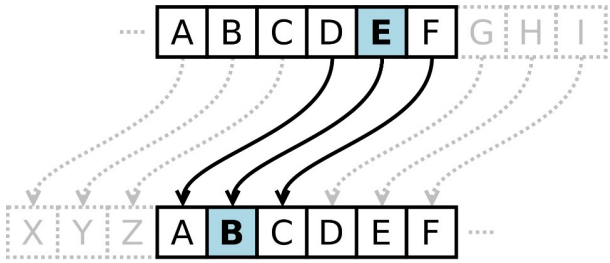


Figure 3: Example of Caesar cipher with the left shift of 3.

Vigenère cipher

Like Caesar cipher, which changes every letter in the message with the different letter. The addition in the Visner cipher is that it changes every letter in the message with the different letter from a different alphabet, i.e. in a different key. The use of the key is done cyclically. After using all the alphabets go back to the first alphabet. The position of each letter in the original message determines which alphabet from the alphabet group of the encryption key is encrypted [1]. In each alphabet key, the order of the letters is different, so that each identical letter in a message will be encrypted to the distinct letter in the cipher, so the frequency of the letters in the original message is not preserved, in contrast to a single alphabetic cipher that

moves all letters at a fixed distance. Therefore, the Visner cipher cannot be cracked with the help of frequency analysis. Figure 4 presents an example of Vigenère cipher.

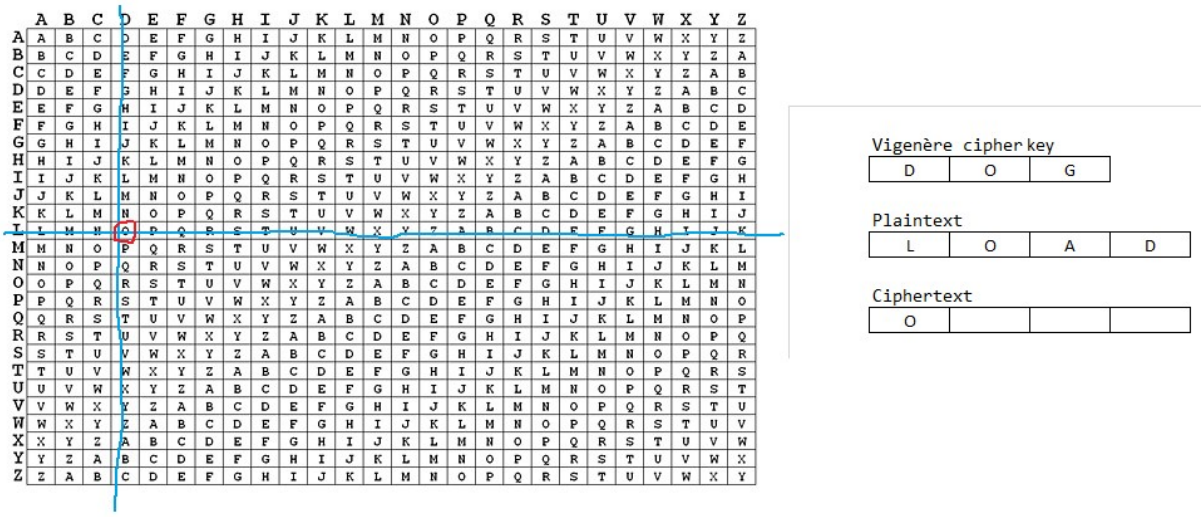


Figure 4: Example of Vigenère cipher

Atbash cipher

Atbash is a mono alphabetic substitution cipher, initially it was used for the encryption of Hebrew alphabets. Because of the standard collating order, the chiper can be changed for use with any known writing system. An example of Atbash cipher is presented in Figure 5.

A	B	C	D	E	F	G	H	I	J	K	L	M
Z	Y	X	W	V	U	T	S	R	Q	P	O	N

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
M	L	K	J	I	H	G	F	E	D	C	B	A

Figure 5: Example of Atbash cipher.

Playfair cipher

This method encrypts pair of letters (digrams or bigrams), inspite of a single letter as in the simple substitution cipher and rather more complex Vigenère cipher systems then in use. This technique is therefore significantly harder to break since the frequency analysis used for simple substitution ciphers does not work with it. The frequency analysis of bigrams is possible, but considerably more difficult. With 600 [3] possible bigrams rather than the 26 possible

monograms (single symbols, usual letters in this context), a considerably larger cipher text is required in order to be useful.

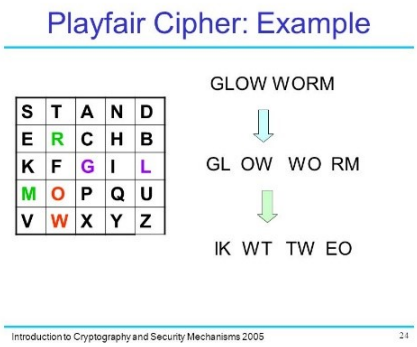


Figure 6: Example of Playfair cipher.

Gronsfeld Cipher

The Count Gronsfeld cipher was created by Gronsfeld cipher (Jose Maximilian van Groensweld and van Bronkhorst). Corresponding to the numbers 0 to 9, it is the same as the Vigener cipher, except that it uses only 10 different ciphers. A Gronsfeld key of 0123 is the same as a Vigenere key of ABCD. The Gronsfeld cipher is strengthened because its key is not a word. But it is weakened because it has just 10 cipher alphabets. Despite its weaknesses, It is Gronsfeld's cipher that became widely used throughout Germany and Europe

		Cleartext																									
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Key	0	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	1	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
	2	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
	3	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
	4	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
	5	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
	6	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
	7	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
	8	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
	9	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I

Ciphertext

Figure 7: Example of Gronsfeld cipher.

Asymmetric encryption

This solves the problem of the number of keys in the case of symmetric cryptography. Here man has only Two keys and the whole world can send him encrypted messages and he can send signed messages and anyone Can verify the signature. On the other hand, it is still unclear how the public keys are distributed. This solves the problem of the number of keys in symmetric

cryptography. In the world man has only two keys and the whole world can send him encrypted messages and he can send signed messages and anyone and also can verify the signature. On the other hand, it is not clear how public keys will be distributed.

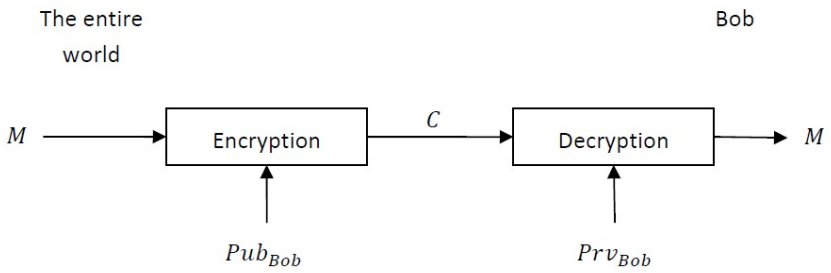


Figure 8: Example of Asymmetric encryption.

RSA Cipher

In the RSA, like any public key system, the encryption key is not secret and is called asymmetric because it is different from the secret key that is kept secret.The asymmetry of the RSA stems from the practical difficulty of decomposing as a factor of a large primitive two-fold absurd number, an open problem in number theory.

```
00 B8 00 02 01 B8 00 15 00 C1 00 F5 00 68 00 93 00 C1 00 78 00
18 01 02 01 FD 00 C1 00 A0 00 93 00 3E 00 93 00 02 01 FD 00 93
00 5F 00 09 00 C1 00 69 00 0F 01 C1 00 2F 00 93 00 EC 00 A0 00
69 00 FD 00 C1 00 F0 00 69 00 3E 00 93 00 15 00 0D 00 38 00 76
00 C1 00 18 01 02 01 FD 00 C1 00 92 00 93 00 A0 00 69 00 76 00
75 00 EC 00 3E 00 69 00 7A 00 C1 00 92 00 F9 00 56 00 02 01 69
00 68 00 74 00 74 00 87 00 DB 00 46 00 46 00 18 01 18 01 18 01
```

Figure 9: Example of RSA Cipher.

Machine learning

Machine learning (ML) is the study of computer algorithms that are automatically enhanced through experience. It can be seen as a subset of artificial intelligence. The machine learning algorithm builds a model based on sample data, called ”training data”, besides, to make predictions or decisions without explicitly programmed. Machine learning algorithms are used in a variety of applications, such as email filtering and computer vision when it is difficult or impossible to develop traditional algorithms to perform the required functions. The machine learning model (called a classifier) is used to solve the employment problem (or similar problems) when given a new instance (a user), predicts a label (whether she is employed or not).

Classifier

Classification is a special case of a hypothesis (nowadays it is often learned by machine learning algorithms). A classifier is a hypothetical or discrete-valued function used to assign (categorical) class labels to specific data points.
A dataset (or dataset) is a collection of data. In the case of table data, a dataset corresponds to one or more database tables, where each column in the table represents a specific variable and each row corresponds to a given record in the processed data.

Train / Test Split

- We must have some labeled data (for example, we need to know whether some users are employed or not) in order to train the classifier.
- To evaluate the performance of our classifier, the data set should be split into train and test sets. Then classifier is trained using a training set and test it on the test set.

Confusion Matrix	Classified as Positive	Classified as Negative
Really Positive	True Positive(TP)	False Negative(FN)
Really Negative	False Positive(FP)	True Negetive(TN)

- Accuracy is the number of correctly predicted data points out of all the data points. Accuracy is calculated as follows.

$$Accuracy = \frac{True}{all} \; or \; \frac{TP + TN}{TP + TN + FN + FP} \tag{1}$$

- Recall means that what proportion of actual positives was identified correctly. Recall is calculatee as follows.

$$Recall = \frac{TruePositive}{ReallyPositive} \; or \; \frac{TruePositive}{TruePositive + FalseNegative} \tag{2}$$

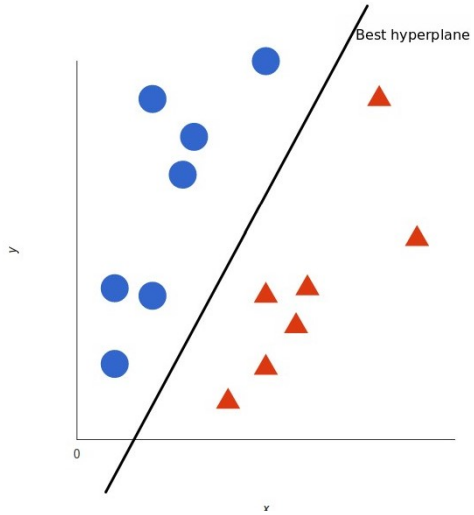
- Precision means that what proportion of positive identifications was actually correct. Precision is calculated as follows.

$$Precision = \frac{TruePositive}{ClassifiedasPositive} \; or \; \frac{TruePositive}{TruePositive + FalsePositive} \tag{3}$$

SVM Machine learning

A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two groups of classification problems. After providing training data labeled in the SVM model for each category, they can categorize the new text. Training examples are represented as vectors in linear space as is customary in this field.

For classification problems, an appropriate classification during the training phase distinguishes between positive and negative training examples as positively as possible. The classification created in SVM is the linear separator which creates as large a space as possible between it and the examples closest to it in the two categories. When a new point appears, the algorithm detects whether it is located inside or outside the line defining the group.



KNN

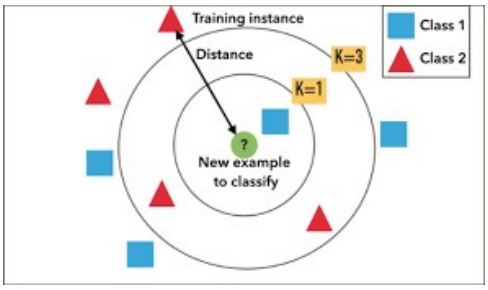
The Near Neighbor Algorithm (or k-NN for short) is a parameter less parameter for local classification and regression.In both cases, the input depends on the closest observation k in the feature space.The k-NN algorithm can be used for classification or regression.

K-NN for classification

According to the input of a new example, the algorithm belongs to the group. Example k is associated with the most common class of immediate neighbors (where k is defined as an integer, usually a small number). The object is associated with the class of the nearest single neighbor when k=1.

K-NN for regression

As a new example, the algorithm returns a sample property value. This value is the average value of the k values of the nearest neighbors. k-NN is a learning algorithm based on examples, or lazy learning, where the function is only approximate locally and all calculations are postponed until it is categorized. The K-NN algorithm is one of the simplest algorithms in machine learning.



Methodology

A large database is needed to train the machine properly. So in order to get the best identification system, we trained our machine on hundreds of thousands of files. The information was always divided into unencrypted files and encrypted files. It is important to note that all the unencrypted information was a lot of real books and text files. In addition, all the encrypted information is encoded in different percentages in order to train the machine properly and come up with an online solution. So that we can save the file as quickly as possible. In order to achieve maximum results in the machine, all the information is divided into 3 types.

1. Only letters of English letters.
2. Only characters and numbers.
3. All types of Basque characters.

Every type of text files have 50% text file that Unencrypted and 50% text file that encrypted by 6 type of encrypted : : Atbash, Autokey, Caesar, Gronsfeld, Playfair, RSA. My Unencrypted took from :

1. Download from this site: `http://www.gutenberg.org/`
2. Divide few big text files to a lot of small text file (200-2200 rows)

The data set was converted into a CSV file with all feature results to train the classifier. We try to get a lot of types of text files with different data and different charts to get the maximum results.

The model was trained with following processing cabalities: CPU – Intel core i7, 3.4 GHz processor, RAM – 16GB, programming language – Python, machine learning library – Scikit-learn [4]

Result and Discussion

All of the results computed are presented in this section. After an in-depth investigation of hundreds of thousands of text files, we came across several repeated patterns in encryption. We can explain to you my solution method, having understood a little of the field of cryptocurrency

and a little of the field of machine learning. Most and all encryption goes at one point to a word that can never be pronounced and at another to a word that doesn't exist in the dictionary. The ideal solution is to go beyond all the words in the text, and if it's a word in the dictionary, keep it appropriate. This solution is recommended for offline work on the files. It takes quite a few weeks for the machine to work to reach the output in order to test all the words in hundreds of thousands of files. Even with partial encryption and the speed of the solution to finding the balance between as accurate answers as possible. We found the combination of efficiency and correctness after doing a lot of experiments.

1. Count the number of words in the text file.
   We need some information to show you the length of my text file. The best way to test it is to count my word in my text.

2. Count if the word starts on small char.
   Because we work on different text files, the file can have 20 rows or 1200 rows.

3. Count if it's a spam word!.
   After reading about cryptographic encryption, we learn how need to look at Suspicious words

4. Count how much popular a word in the text.
   We took the popular 100 English words to an array and count how much popular word in text (this feature based on feature 5 what we will explain next slide)

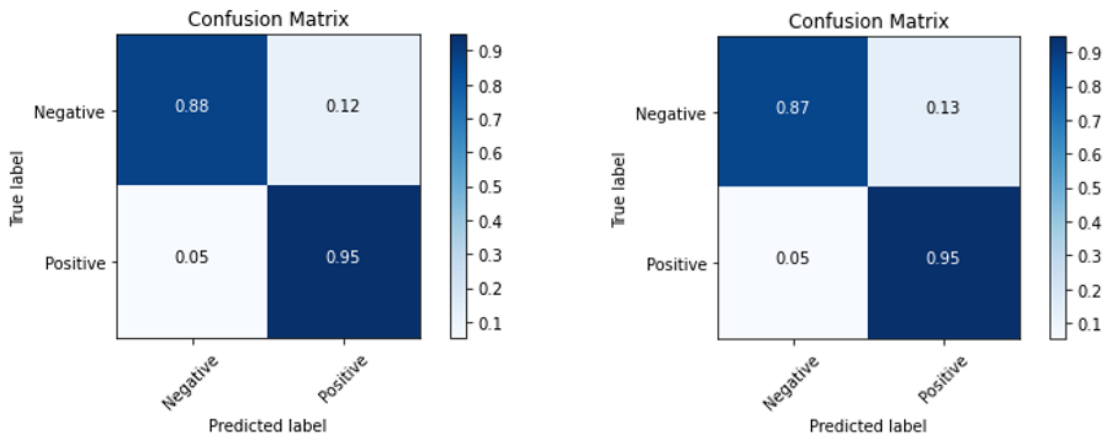
5. Check if the word is in English or no.
   Basically, this feature is enough in a descriptive word

6. Count how many line on text.
   This feature is to check how many lines in the text

Table 1 presents a detail comparison of classification accuracies. It is observable that both classifiers performed well on the proposed model. Figure 10 shows the confusion matrices, that presents the classification accuracies between the classes with both classifiers: (a) KNN and (b) SVM. We also computed the results with each encryption type, Figure 11 shows the bar chart that presents classification accurcies with each encryption type. It is observable that Caesar cipher detects with 100% accuracy.

Table 1: Results obtained by SVM and KNN.

Model	**Precision**	**Recall**	**F1-Score**
SVM	91%	91%	91%
KNN	92%	91%	91%



(a) (b)

Figure 10: Confusion matrices presents the classification accuracies: (a) KNN and (b) SVM.

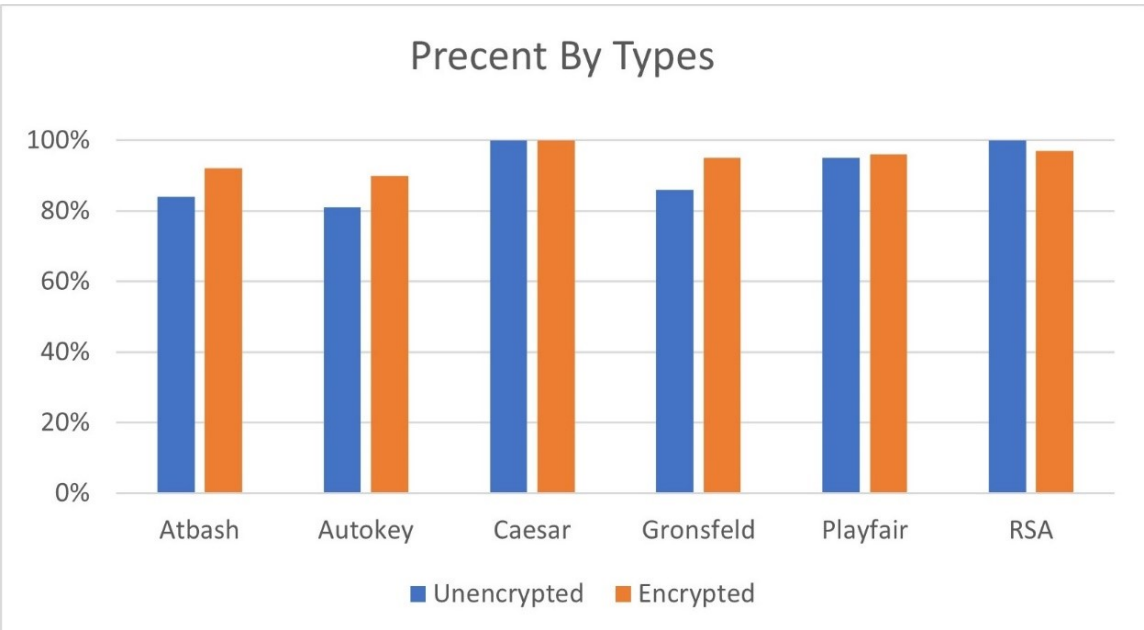


Figure 11: The figure shows that the accuracy percentages separate by encrypted type. As a result, Caesar cipher detects with 100% accuracy.

References

[1] A. Azmoodeh, A. Dehghantanha, M. Conti, and K.-K. R. Choo, "Detecting crypto-ransomware in iot networks based on energy consumption footprint," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 4, pp. 1141–1152, 2018.

[2] N. Scaife, H. Carter, P. Traynor, and K. R. Butler, "Cryptolock (and drop it): stopping ransomware attacks on user data," in *2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS)*, pp. 303–312, IEEE, 2016.

[3] M. M. Ahmadian and H. R. Shahriari, "2entfox: A framework for high survivable ransomwares detection," in *2016 13th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC)*, pp. 79–84, IEEE, 2016.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.