International Journal of Future Generation Communication and Networking
Vol. 15, No. 1, (2022), pp. 168–180

XGBOOST as A Classifier for Security Level Detection of Cryptosystem

Anyam Mohan[1], Dr. S. Raj Anand[2]

[1]PG Scholar, Dept. of CSE, VEMU Institute of Technology, P.Kothakota, Chittoor.

[2] Professor, Dept. of CSE, VEMU Institute of Technology, P.Kothakota, Chittoor.

anyammohan332@gamil.com 1 drrajanands@gmail.com2

Abstract

By now, we have most likely heard of, or used, Zoom, the video conferencing service. Due to the coronavirus pandemic, Zoom has experienced an enormous spike in use over the past few months.Unfortunately, that same ease of use seems to have led to a variety of security and privacy issues.In short, Zoom's meeting encryption exhibited less than "end-to-end" fortitude. In line with their privacy practices, the image, video and audio content during a Zoom meeting would remain private from any outsider (i.e., hackers). On this note, we can clearly tell that, though, several proposed encryption algorithms proven failed by allowing major vulnerabilities against important data. To reduce such vulnerabilities, it's very important that which encryption algorithm is being used to protect the data. And we've to know which algorithm is suitable for that specific image, video or audio encryption along with the accuracy of encryption. Here for to full fill this need, we propose as security level detection approach for finding "strong, medium and weak" image encryption algorithms by incorporating a XGBoost Classifier Machine learning technique to reduce the effort of time complexity on detecting the appropriate encryption method.

Keywords: - XGBoost Classifier, Zoom, image encryption, vulnerabilities

I. Introduction

Due to the exponential increase in transmissions of multimedia data over insecure channels (mostly the Internet), security has become a much-in-demand area of research. To protect data from eavesdroppers and unauthorized users, many researchers have turned to developing new encryption algorithms [1]– [5].

Two factors are crucial, when encrypting any digital images knows as diffusion and confusion (also called scrambling). In [6], Claud Shannon proposed a theory that cryptosystem contains confusion and diffusion mechanisms, may be considered a secure cryptosystem. With digital images, the scrambling process can be performed directly either on pixels or else on rows and columns, whereas diffusion changes the originalpixel values.

In Zoom and other platforms where multimedia is transmitting over internet are need to have end-to-end encryption to protect user data against privacy policy. Major problems facing in cyber-attacks now a days are through malware mails also called phish-ing mails and over internet ads on webpages.

Here, for an example, we have used Lorenz encoding technique to demonstrate the encryption output. Where we are evident that after performing encryption also, we're seeing the apple colours are getting recognized by human eye. Strong encrypti-on is one which will encrypt the plain image completely, enabling it to resist attacks against its integrity, confidentiality, and availability as well as visibility.

1(a): Plan apple image 1(b): Encrypted image with Lorenz encoding

On taking one example from recent days, Hackers exploit NASA’s famous deep space image to attack computers. A newly-discovered hacking campaign is exploiting an image from the James Webb Telescope to infect targets with malware.“Initial infection begins with a phishing email containing a Microsoft Office attachment. The document includes an external reference hidden inside the document’s metadata which downloads a malicious template file,” said the researchers. When the document is opened, the malicious template file is downloaded and saved on the system. Finally, the script downloads a JPEG image that shows the James Webb Telescope deep field image.

For to overcome these cyber-attacks, researchers proposed many models with SVM classifier (Machine learning model). But in that technique the accuracy is less and it will take more time.
But we’re proposing a security level detection approach for finding “strong, medium and weak” image encryption algorithms by incorporating a XGBoost classifier (Machine learning model) based on the security parameters such as entropy, homogeneity, contrast, correlation, energy, PSNR and MSE. Which will boost the accuracy and the processing time.

As we are targeting those encryption algorithms, which are used to encrypt the 8–bit images. For the 8–bit images, the maximum entropy cannot be exceeded by 8. Likewise, for the binary images, the maximum entropy that can be obtained is 2. So, in the case of 8-bit images, we have divided the whole entropy interval for 8-bit images into three intervals. The range of the whole interval is 0 to 8.
The average entropy value of any plain image may vary from 7.600 to 7.700. Whereas, an enciphered image encrypted generated using a weak encryption algorithm such as a Lorenz encoding may produce the average entropy value between 7.9503 to 7.9799. While for an acceptable and strong encryption algorithm, the average entropy value may vary from 7.9800 to 7.9900 and 7.9901 to 8.000 respectively. Similarly, the values for other security parameters may vary accordingly.

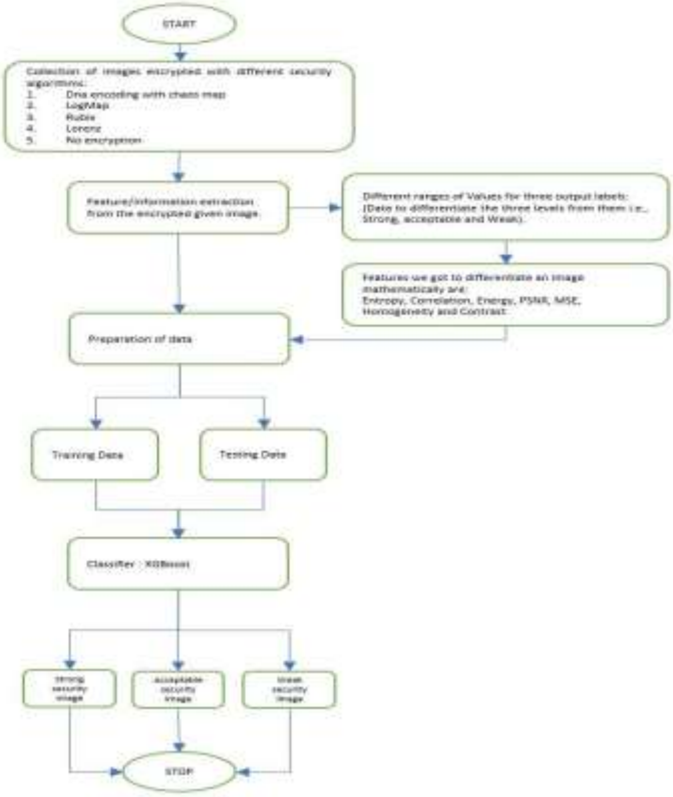
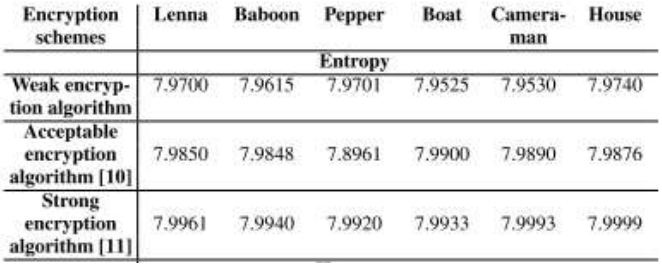

START
Collection of images encrypted with different security algorithms:
1. Dna encoding with chaos map
2. LogMap
3. Rubix
4. Lorenz
5. No encryption
Feature/information extraction from the encrypted given image.
Different ranges of Values for three output labels: (Data to differentiate the three levels from them i.e., Strong, acceptable and Weak).
Features we got to differentiate an image mathematically are: Entropy, Correlation, Energy, PSNR, MSE, Homogeneity and Contrast
Preparation of data
Training Data
Testing Data
Classifier : XGBoost
strong security image
acceptable security image
weak security image
STOP

Fig: Workflow diagram

Rules for classification: To classify the encryption algorithms into three different categories (strong, medium and weak), the following are the rules must follow by the proposed model. For the classification of each category, the decision will be based on the values of the security parameter.

- We have divided the range of each of the parameters into three intervals defined for weak, acceptable and strong security. For the weak security level, below 50 percent feature values must lie in the acceptable interval values.
- For acceptable security, at least 65 percent feature values must lie in the acceptable interval values.
- For strong security, more than 80 percent features values must lie in the acceptable interval values.

Table 1: Statical values of entropy for different encrypted images.



Encryption schemes	Lenna	Baboon	Pepper	Boat	Camera-man	House
			Entropy			
Weak encryption algorithm	7.9700	7.9615	7.9701	7.9525	7.9530	7.9740
Acceptable encryption algorithm [10]	7.9850	7.9848	7.8961	7.9900	7.9890	7.9876
Strong encryption algorithm [11]	7.9961	7.9940	7.9920	7.9933	7.9993	7.9999

II Literature Review

A number of encryption algorithms have been proposed as means of securing images before transmission. Encryption algorithms may develop based on chaos or transformation methods, such as discrete wavelet transformation, discrete cosine transformation and discrete Fourier transformation [12]–[17]. These are just some of the many image encryption schemes that have been proposed in recent years, though. Further details of each type are provided below: In [18], a cosine transformation and chaos-based image encryption algorithm was proposed. Here, three different chaotic maps were used instead of a single chaotic system. The proposition of using more than one chaotic map was to create more complexity in the overall algorithm, thus enabling it to exhibit more complicated and dynamic behavior. To enhance the security of the encryption algorithm, Kaur et. al proposed a new optical image encryption scheme based on a chaotic in [19] which proved capable of generating the vectors of multiple orders using a piece-wise linear chaotic map (PWLCM) [20]. For a fast image encryption, Khan et al. proposed a chaos-based selective image encryption scheme in [21]. Although selective encryption schemes work well for real-time applications where fast encryption is required, they are not suitable for text encryption, where every individual single bit must be encrypted in order for the data to be properly concealed. These algorithms achieved efficient encryption, as demonstrated by the statistical analysis; however, these results were not enough to show the security level of the proposed work. More analysis would be needed to show a better assessment of that particular encryption algorithm. Although the chaos has an ability to generate random number, Nardo et al. explained the limitations of chaos-based encryption schemes in [22], claiming that these types of encryption algorithms are implemented on a finite precision computer, causing dynamic degradation that makes the chaos-based encryption insecure. To encrypt plain images, the authors used a finite precision error, which was generated by the implementation of chaos-based systems using different interval delays. Explaining few moew limitations of chaos, the authors in [23] claimed that chaos-based communication systems are not secure enough because they depend on initial values, meaning that their security can be broken by identifying those initial values. To enhanced the security of the chaos-based crptosystem, in our previous work, a bit-plane extraction method is incorporated to propose a new image encryption technique based on multiple chaotic systems [24]. The main aim of the proposed technique was to reduce the necessary processing time while also increasing the available concealment. In [10], a chaotic logistic map (CLM) [25]-based image encryption algorithm is proposed. In this work, the author addressed the issues of a using single substitution box (S-box) encryption by using multiple S-box image encryption in which the selection of a particular S-box depends on the random values generated by the CLM. In chaos-based image encryption, S-boxes are a frequent component, given their powerful, nonlinear provision of a diffusion source. S-boxes thus play a vital role in transforming the original data into an encoded format. Because the strength of chaos-based encryption algorithms depends on the robustness of the S-box, this component must be strong enough to resist statistical attacks. The development of strong S-boxes is a critical research area for security professionals.

To overcome the issues of using weak S-box, we previously proposed a CLM-based methodology capable of creating a new S-box in [26]. The values of the S-box thus generated may vary by a slight change in the initial values of CLM. Apart from the gray scale image encryption, a color image is even more challenging than the encryption of a gray image. This is because with color image encryption, all three channels (R, G, B) must be encrypted. In [27], a color image encryption technique is proposed that utilizes a hybrid chaotic system. The authors used the phenomenon of confusion for the encryption of each R, G, and B component separately and then a mitochondrial DNA sequence was used to diffuse the confused components. Each of the encryption algorithms explained above has a different level of security: i.e., some are strong, some are acceptable, and some are weak. Which category an algorithm falls into depends on how complex its mathematical structure is.

III Xgboost As A Classifierproposed Model For Security Level Detection Of Cryptosytem

To get started with xgboost, just install it either with pip or conda:

```
# pip
pip install xgboost
# conda
conda install -c conda-forge xgboost
```

After installation, you can import it under its stan-dard alias -xgb. For classification problems, the library provides XGBClassifier class:

```
from tkinterimport FIRST
import numpyas np
import pandas as pd
from sklearn.model_selectionimport train_test_split
import pickle
import xgboostas xgb
```

Fortunately, the classifier follows the familiar fit-predict pattern of sklearn meaning we can freely use it as any sklearn model.

Before we train the classifier, let’s preprocess the data and divide it into train and test sets:

```
data = pd.read_csv('dataset.csv')
data.drop_duplicates(subset=['Entropy','Energy','Contrast','Correlation','Homogeneity','MSE','PSNR','
Security'],keep=FIRST, inplace=True)
print("Shape: ",data.head())
x = data.iloc[:, 1:-1]
y = data.iloc[:, -1]

x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.3, random_state=5)
```

Now, we fit the classifier with default parameters and evaluate its performance:

```
from sklearn.preprocessingimport StandardScaler

st_x = StandardScaler()
x_train = st_x.fit_transform(x_train)
x_test = st_x.transform(x_test)

from sklearn.svmimport SVC
classifier = SVC(kernel='rbf', random_state=1)
classifier.fit(x_train, y_train)
y_pred = classifier.predict(x_test)

print("----------SVM--------------")
import sklearn.metricsas mt
```

```
cm = mt.confusion_matrix(y_true=y_test, y_pred=y_pred)
print("Confusion Matrix")
print(cm)
acc = mt.classification_report(y_true=y_test, y_pred=y_pred)
print(acc)



print("---------XGB-------------")


xb = xgb.XGBClassifier()
clf = xb.fit(x_train, y_train)
y_pred = clf.predict(x_test)

cm = mt.confusion_matrix(y_true=y_test, y_pred=y_pred)
print("Confusion Matrix")
print(cm)
acc = mt.classification_report(y_true=y_test, y_pred=y_pred)
print(acc)
```

XGBoost provide an 99% accuracy.

Dataset:

In order to detect the security level of a given algorithm, the following steps should be performed:

- Take a big collection of data from different cipher images generated using various encryption algorithms [10], [21], [29]–[33]. The cipher images data taken is given below:

Table 2: Dataset values taken for to train the XGBoost Classifier Model.

A. SECURITY PARAMETERS AS FEATURES

1) CONTRAST

$$Cont = \sum |x-y|^2 z(x,y)$$

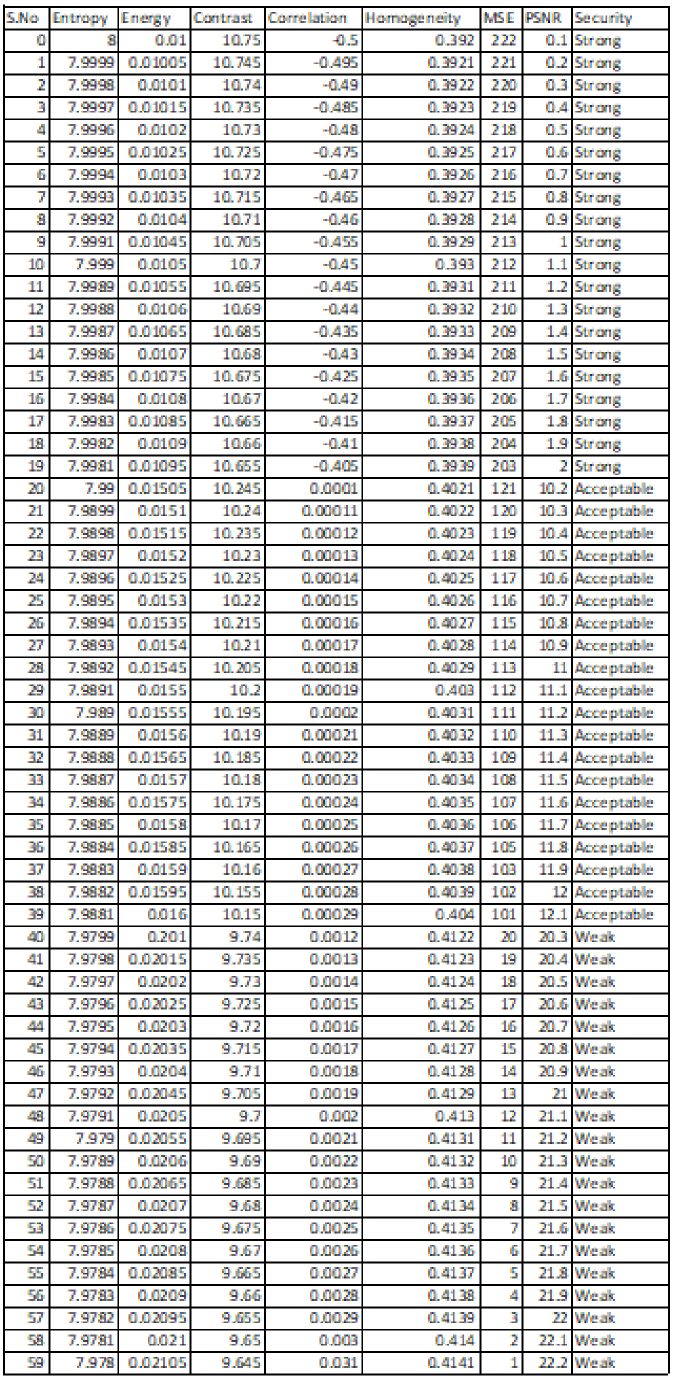
2) ENTROPY

$$Entropy = \sum_{d=1}^{M} p(s_m) log_2(p(s_m))$$

3) ENERGY

$$Energy = \sum_{K=1}^{L} im(x,y)^2$$

Table 2: Dataset values taken for to train XGBoost.



S.No	Entropy	Energy	Contrast	Correlation	Homogeneity	MSE	PSNR	Security
0	8	0.01	10.75	-0.5	0.392	222	0.1	Strong
1	7.9999	0.01005	10.745	-0.495	0.3921	221	0.2	Strong
2	7.9998	0.0101	10.74	-0.49	0.3922	220	0.3	Strong
3	7.9997	0.01015	10.735	-0.485	0.3923	219	0.4	Strong
4	7.9996	0.0102	10.73	-0.48	0.3924	218	0.5	Strong
5	7.9995	0.01025	10.725	-0.475	0.3925	217	0.6	Strong
6	7.9994	0.0103	10.72	-0.47	0.3926	216	0.7	Strong
7	7.9993	0.01035	10.715	-0.465	0.3927	215	0.8	Strong
8	7.9992	0.0104	10.71	-0.46	0.3928	214	0.9	Strong
9	7.9991	0.01045	10.705	-0.455	0.3929	213	1	Strong
10	7.999	0.0105	10.7	-0.45	0.393	212	1.1	Strong
11	7.9989	0.01055	10.695	-0.445	0.3931	211	1.2	Strong
12	7.9988	0.0106	10.69	-0.44	0.3932	210	1.3	Strong
13	7.9987	0.01065	10.685	-0.435	0.3933	209	1.4	Strong
14	7.9986	0.0107	10.68	-0.43	0.3934	208	1.5	Strong
15	7.9985	0.01075	10.675	-0.425	0.3935	207	1.6	Strong
16	7.9984	0.0108	10.67	-0.42	0.3936	206	1.7	Strong
17	7.9983	0.01085	10.665	-0.415	0.3937	205	1.8	Strong
18	7.9982	0.0109	10.66	-0.41	0.3938	204	1.9	Strong
19	7.9981	0.01095	10.655	-0.405	0.3939	203	2	Strong
20	7.99	0.01505	10.245	0.0001	0.4021	121	10.2	Acceptable
21	7.9899	0.0151	10.24	0.00011	0.4022	120	10.3	Acceptable
22	7.9898	0.01515	10.235	0.00012	0.4023	119	10.4	Acceptable
23	7.9897	0.0152	10.23	0.00013	0.4024	118	10.5	Acceptable
24	7.9896	0.01525	10.225	0.00014	0.4025	117	10.6	Acceptable
25	7.9895	0.0153	10.22	0.00015	0.4026	116	10.7	Acceptable
26	7.9894	0.01535	10.215	0.00016	0.4027	115	10.8	Acceptable
27	7.9893	0.0154	10.21	0.00017	0.4028	114	10.9	Acceptable
28	7.9892	0.01545	10.205	0.00018	0.4029	113	11	Acceptable
29	7.9891	0.0155	10.2	0.00019	0.403	112	11.1	Acceptable
30	7.989	0.01555	10.195	0.0002	0.4031	111	11.2	Acceptable
31	7.9889	0.0156	10.19	0.00021	0.4032	110	11.3	Acceptable
32	7.9888	0.01565	10.185	0.00022	0.4033	109	11.4	Acceptable
33	7.9887	0.0157	10.18	0.00023	0.4034	108	11.5	Acceptable
34	7.9886	0.01575	10.175	0.00024	0.4035	107	11.6	Acceptable
35	7.9885	0.0158	10.17	0.00025	0.4036	106	11.7	Acceptable
36	7.9884	0.01585	10.165	0.00026	0.4037	105	11.8	Acceptable
37	7.9883	0.0159	10.16	0.00027	0.4038	103	11.9	Acceptable
38	7.9882	0.01595	10.155	0.00028	0.4039	102	12	Acceptable
39	7.9881	0.016	10.15	0.00029	0.404	101	12.1	Acceptable
40	7.9799	0.201	9.74	0.0012	0.4122	20	20.3	Weak
41	7.9798	0.02015	9.735	0.0013	0.4123	19	20.4	Weak
42	7.9797	0.0202	9.73	0.0014	0.4124	18	20.5	Weak
43	7.9796	0.02025	9.725	0.0015	0.4125	17	20.6	Weak
44	7.9795	0.0203	9.72	0.0016	0.4126	16	20.7	Weak
45	7.9794	0.02035	9.715	0.0017	0.4127	15	20.8	Weak
46	7.9793	0.0204	9.71	0.0018	0.4128	14	20.9	Weak
47	7.9792	0.02045	9.705	0.0019	0.4129	13	21	Weak
48	7.9791	0.0205	9.7	0.002	0.413	12	21.1	Weak
49	7.979	0.02055	9.695	0.0021	0.4131	11	21.2	Weak
50	7.9789	0.0206	9.69	0.0022	0.4132	10	21.3	Weak
51	7.9788	0.02065	9.685	0.0023	0.4133	9	21.4	Weak
52	7.9787	0.0207	9.68	0.0024	0.4134	8	21.5	Weak
53	7.9786	0.02075	9.675	0.0025	0.4135	7	21.6	Weak
54	7.9785	0.0208	9.67	0.0026	0.4136	6	21.7	Weak
55	7.9784	0.02085	9.665	0.0027	0.4137	5	21.8	Weak
56	7.9783	0.0209	9.66	0.0028	0.4138	4	21.9	Weak
57	7.9782	0.02095	9.655	0.0029	0.4139	3	22	Weak
58	7.9781	0.021	9.65	0.003	0.414	2	22.1	Weak
59	7.978	0.02105	9.645	0.031	0.4141	1	22.2	Weak

4) Correlation

$$\mu_{ab} = \frac{E[a - E(a)][y - E(b)]}{\sqrt{D(a)}\sqrt{D(b)}}$$

Where,

$$\mathrm{E(a)} = \frac{1}{M}\sum\nolimits_{i=1}^{M} a_i$$

5) HOMOGENEITY

$$\sum_{a}\sum_{b}\frac{P(a,b)}{1+|a-b|}$$

6) PEAK SIGNAL TO NOISE RATIO (PSNR) AND MEAN SQUARE ERROR (MSE)

$$PSNR = 20log_{10}(\frac{max_{val}}{\sqrt{MSE}})$$

where $\max_{val}$ signifies the highest pixel value present in the plain image

$$MSE = \frac{1}{XY}\sum_{a=1}^{X}\sum_{b=1}^{Y}(P_{im}(a,b) - C_{im}(a,b))$$

- The dataset is created using the intervals explained above. Once the dataset is created, a portion of it will be separated for training purposes while the rest is used for testing.
- After the training and testing stages, we will extract the features from another cipher image in order to attempt the prediction of the security level achievable by the encryption algorithm through which the cipher image is generated.
- Finally, to evaluate our proposed model, we will test its accuracy, F1 score, recall, and precision.

IV. Statistical Analysis of the Proposed Model

To evaluate the performance of the proposed model, we have done some experimental analysis, as outlined below.

A. CONFUSION MATRIX

The confusion matrix is a two-dimensional array that can be utilized to find accuracy, recall. and precision.

An explanation of these four terms according to the proposed model is given below.

1) TRUE POSITIVES

When the system predicts ‘‘strong security’’ while the real output was also ‘‘strong security’’.

2) TRUE NEGATIVES

When the system predicts ‘‘acceptable security’’ while the real output was also ‘‘acceptable security’’.

3) FALSE POSITIVES

When the system predicts ‘‘strong security’’ while the real output was ‘‘acceptable or weak security’’.

4) FALSE NEGATIVES

When the system predicts ‘‘acceptable security’’ or ‘‘weak security’’ while the real output was ‘‘strong security’’.

Or

When the system predicts ‘‘weak security’’ while the real output was ‘‘acceptable security’’

By using the confusion matrix, accuracy can be expressed as:

- Accuracy = Addition of all the values of first diagonal / total number of samples

B) CLASSIFICATION ACCURACY

The accuracy of this system reveals the information about how many correct predictions have been made by the model. The more correct predictions made, the higher the resulting accuracy. This classification accuracy can be measured as:

- Classification accuracy = No. of correct predictions / Total number of predictions

C. PRECISION AND RECALL

Precision is the ratio between the true positive predicted observations and the total number of positive predicted observations. Mathematically, this can be expressed as:

- Precision = T.P / (T.P + F.P)

In the case of our proposed work, the precision will be:

- Precision = T.P / (T.P + $(F.P)_{(1)}$ + $(F.P)_{(2)}$)

Recall refers to the sensitivity of the model. The greater the recall score, the more sensitive the model will be. In other words, this expresses the ratio of true positive observation and the total number of true positive and false negative observations. Mathematically, recall can be calculated as:

- Recall = T.P / (T.P + F.N)

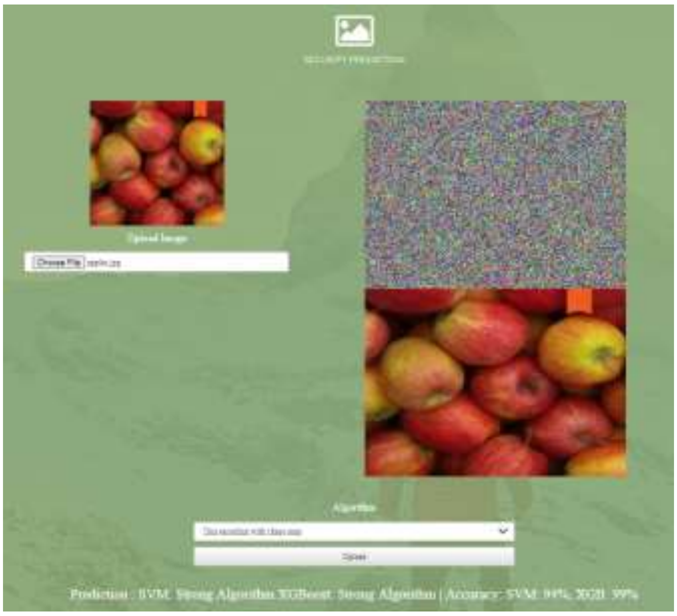
D. F1 SCORE

Accuracy and F1 score both are important metrics when evaluating the performance of machine learning models. Accuracy is important when true positive and true negative samples are more valuable, while the F1 score is important when false positive and false negative samples are more important. F1 score can be calculated as:

$$\begin{aligned}\text{F1 Score} &= \left[\frac{(\text{Recall})^{-1} + (\text{Precision})^{-1}}{2}\right]^{-1} \\ &= 2\left(\frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}\right)\end{aligned}$$

V. Implementation and Comparisions

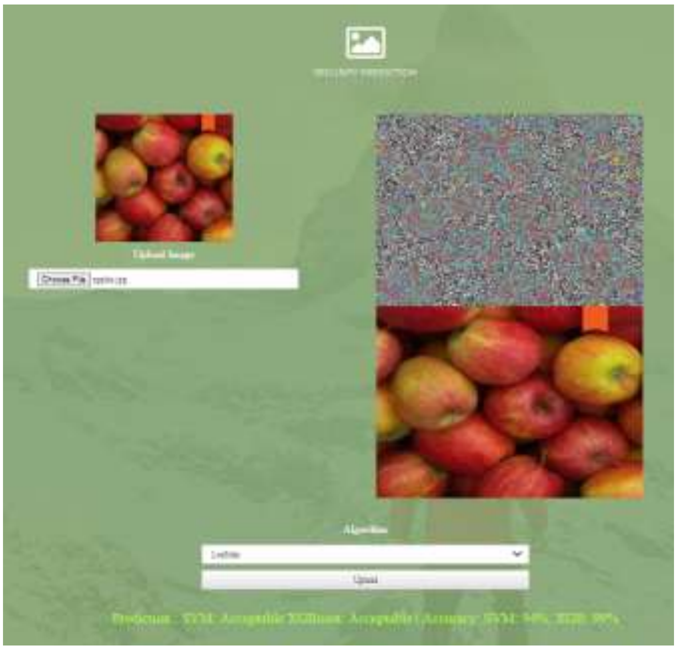
1) Uploading an unencrypted “apples” image with DNA encoding technique then finding it’s level of encryption happen with the DNA encoding technique:



Prediction : SVM: Strong Algorithm XGBoost: Strong Algorithm | Accuracy: SVM: 94%, XGB: 99%

- Prediction/Result: Strong Algorithm

2) Uploading an unencrypted "apples" image with LogMapencoding technique then finding it's level of encryption happen with the LogMapencoding technique:

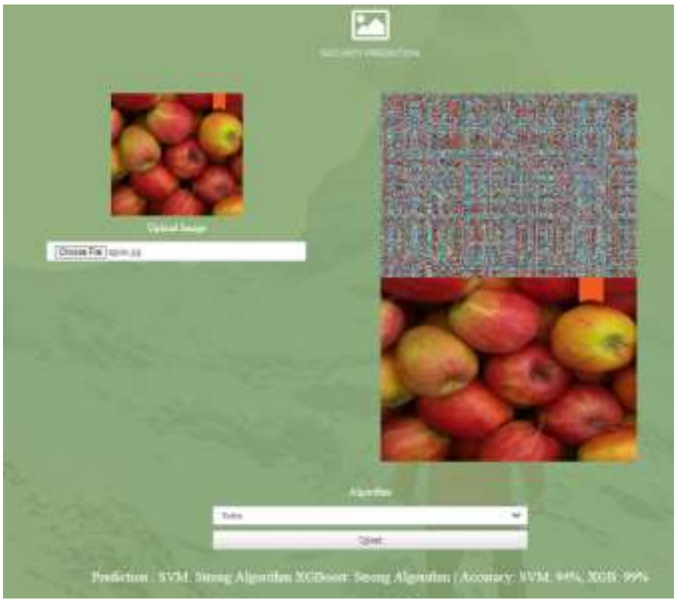


Prediction : SVM: Acceptable XGBoost: Acceptable | Accuracy: SVM: 94%, XGB: 99%

- Prediction/Result: Acceptable Algorithm

3) Uploading an unencrypted "apples" image with Rubixencoding technique then finding it's level of encryption happen with the Rubixencoding technique:

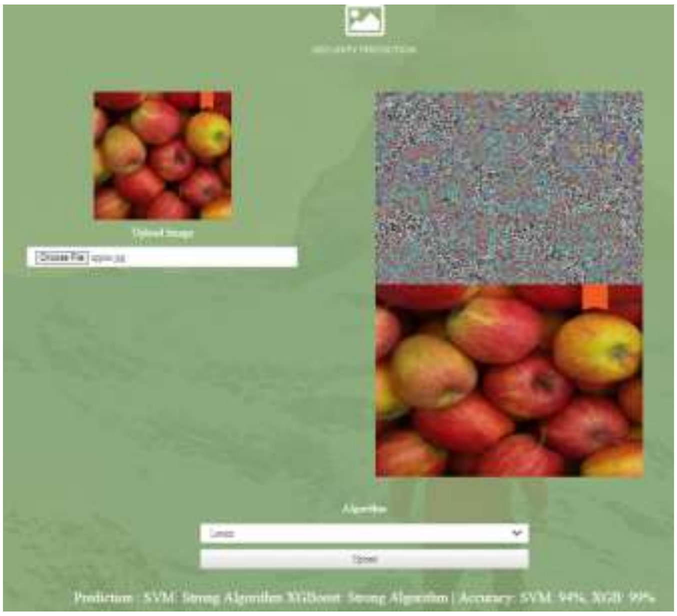
ISSN: 2233-7857IJFGCN



Prediction : SVM: Strong Algorithm XGBoost: Strong Algorithm | Accuracy: SVM: 94%, XGB: 99%

- Prediction/Result: Strong Algorithm

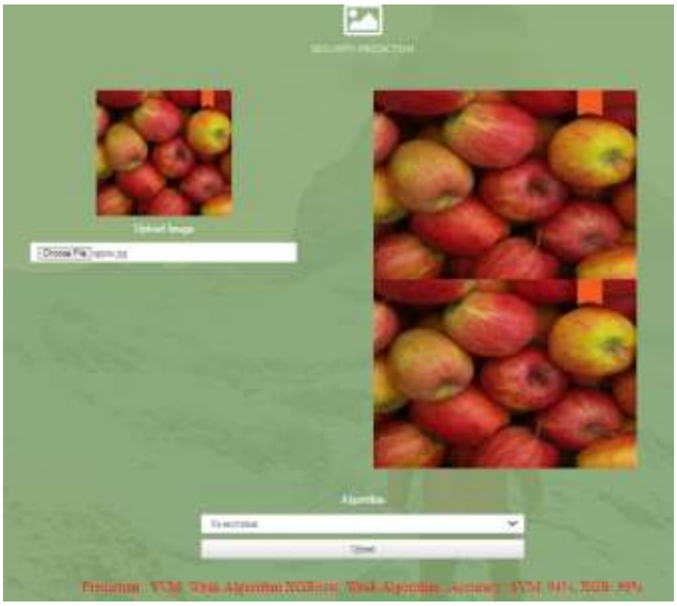
4) Uploading an unencrypted “apples” image with Lorenz encoding technique then finding it’s level of encryption happen with the Lorenz encoding technique:



Prediction : SVM: Strong Algorithm XGBoost: Strong Algorithm | Accuracy: SVM: 94%, XGB: 99%

- Prediction/Result: Strong Algorithm

5) Uploading an unencrypted “apples” image with No/without encoding technique then finding it’s level of encryption happen with the No/without encoding technique:



Prediction : SVM: Weak Algorithm XGBoost: Weak Algorithm Accuracy : SVM: 94% XGB: 99%

- Prediction/Result: Weak Algorithm

VI. Conclusion and Future Enhancements

In this article, to enhance the security and privacy issues of users who are using internet for exchange of image, audio & video through Zoom and other social platforms, we have developed and proposed a model that can detect the security level of various encryption schemes quickly and accurately. We began by creating a dataset and incorporating the security parameters common to various encryption schemes as features. To prepare a dataset, we have divided the values of all features into three intervals i.e., strong, acceptable, and weak, that describe the resulting security levels. Next, the different encryption schemes are tested on our proposed model in order to detect the level of security each one offers. We can also detect the security level of these encryption schemes manually by determining the statistical values of each one. With traditional testing methods, this process takes a great deal of time to accomplish but with our proposed model, testing can be achieved within a few seconds. To conclude, we also tested our proposed model using different experiments to evaluate its performance, and we found that it produces 99% correct predictions at much faster speeds than other models currently available.

In the future work, the use of deep learning techniques to detect the security level of cryptosystems will be investigated [37], [38].

References

[1] I. Hussain, A. Anees, A. H. Alkhaldi, M. Aslam, N. Siddiqui, and R. Ahmed, ‘‘Image encryption based on Chebyshev chaotic map and S8 S-boxes,’’ *Optica Applicata,* vol. 49, no. 2, pp. 317–330, 2019.

[2] A. Anees, I. Hussain, A. Algarni, and M. Aslam, ‘‘A robust watermarking scheme for online multimedia copyright protection using new chaotic map,’’ *Secur. Commun. Netw.*, vol. 2018, pp. 1–20, Jun. 2018.

[3] A. Shafique and J. Ahmed, ‘‘Dynamic substitution based encryption algorithm for highly correlated data,’’ *Multidimensional Syst. Signal Process.*, May 2020.

[4] F. Ahmed, A. Anees, V. U. Abbas, and M. Y. Siyal, ‘‘A noisy channel tolerantimageencryptionscheme,’’*WirelessPers.Commun.*,vol.77,no.4, pp. 2771–2791, Aug. 2014.

[5] M. A. B. Farah, R. Guesmi, A. Kachouri, and M. Samet, ‘‘A novel chaos based optical image encryption using fractional Fourier transform and DNA sequence operation,’’ *Opt. Laser Technol.*, vol. 121, Jan. 2020, Art. no. 105777.

[6] C. E. Shannon, ‘‘Communication in the presence of noise,’’ *Proc. IEEE*, vol. 72, no. 9, pp. 1192–1201, Sep. 1984.

[7] S. Heron, ‘‘Advanced encryption standard (AES),’’ *Netw. Secur.*, vol. 2009, no. 12, pp. 8–12, Dec. 2009.

[8] H. Liu, A. Kadir, and X. Sun, ‘‘Chaos-based fast colour image encryption scheme with true random number keys from environmental noise,’’ *IET Image Process.*, vol. 11, no. 5, pp. 324–332, Apr. 2017.

[9] Y.-L. Lee and W.-H. Tsai, ‘‘A new secure image transmission technique via secret-fragment-visible mosaic images by nearly reversible color transformations,’’ *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 4, pp. 695–703, Apr. 2014.

[10] A. Anees, A. M. Siddiqui, and F. Ahmed, ‘‘Chaotic substitution for highly autocorrelated data in encryption algorithm,’’ *Commun. Nonlinear Sci. Numer. Simul.*, vol. 19, no. 9, pp. 3106–3118, Sep. 2014.

[11] L. Liu, Y. Lei, and D. Wang, ‘‘A fast chaotic image encryption scheme with simultaneous permutation-diffusion operation,’’ *IEEE Access*, vol. 8, pp. 27361–27374, 2020.

[12] M. Khalili and D. Asatryan, ‘‘Colour spaces effects on improved discrete wavelet transform-based digital image watermarking using Arnold transform map,’’ *IET Signal Process.*, vol. 7, no. 3, pp. 177–187, May 2013.

[13] L. Zhang, J. Wu, and N. Zhou, ‘‘Image encryption with discrete fractional cosine transform and chaos,’’ in *Proc. 5th Int. Conf. Inf. Assurance Secur.*, vol. 2, 2009, pp. 61–64.

[14] M. Zhang, X.-J. Tong, J. Liu, Z. Wang, J. Liu, B. Liu, and J. Ma, ‘‘Image compression and encryption scheme based on compressive sensing and Fourier transform,’’ *IEEE Access*, vol. 8, pp. 40838–40849, 2020.

[15] J.S.Khan,W.Boulila,J.Ahmad,S.Rubaiee,A.U.Rehman,R. Alroobaea, and W. J. Buchanan, ‘‘DNA and plaintext dependent chaotic visual selective image encryption,’’ *IEEE Access*, vol. 8, pp. 159732–159744, 2020.

[16] A. Qayyum, J. Ahmad, W. Boulila, S. Rubaiee, Arshad, F. Masood, F. Khan, and W. J. Buchanan, ‘‘Chaos-based confusion and diffusion of image pixels using dynamic substitution,’’ *IEEE Access*, vol. 8, pp. 140876–140895, 2020.

[17] F. Masood, W. Boulila, J. Ahmad, Arshad, S. Sankar, S.sssRubaiee, and W. J. Buchanan, ‘‘A novel privacy approach of digital aerial images based on mersenne twister method with DNA genetic encoding and chaos,’’ *Remote Sens.*, vol. 12, no. 11, p. 1893, Jun. 2020.