Machine Learning Empowered: Support Vector Machine-Based Selection of Encryption Techniques for Digital Image Security Levels

Nitin Shivsharan, Mandar Tari
Department of Computer Engineering, SSPM's College of Engineering, Kankvali, India
E-mail: shivsharan.nitin@gmail.com, mandartari9109@gmail.com

Keywords: Machine learning, cryptosystem, image encryption, support vector machine, performance evaluation, confusion matrix

Received: June 3, 2024

Recent advancements in multimedia systems have increased the demand for strong digital image security mechanisms. Traditional cryptosystem evaluations, dependent on manual statistical analysis, are computationally exhaustive and not scalable. This study proposes a machine learning-based framework, such as a Support Vector Machine (SVM) classifier, for the categorization of image encryption levels into three discrete classes: Strong, Acceptable, and Weak. The model is trained using statistical descriptors such as Peak Signal to Noise Ratio (PSNR), entropy, Mean Square Error (MSE), energy, correlation, homogeneity, and contrast extracted from encrypted image datasets. Feature normalization techniques, StandardScaler, have been used to ensure balanced input contributions. The proposed system, i.e., SVM with a Radial Basis Function (RBF) kernel, outperforms other kernels. The performance of the proposed model shows an average classification accuracy of 98%, precision up to 100%, and an F1-score of 97%. A web-based interface developed using Django integrates the model, enabling real-time analysis and visualization, making the proposed system a scalable solution for cryptographic strength evaluation in image security applications.

Povzetek: Raziskava je razvila SVM-okvir za razvrščanje varnostnih stopenj slikovne enkripcije, ki s statističnimi značilnicami dosega visoko točnost in omogoča praktično spletno analizo v realnem času.

1 Introduction

Due to the rapid growth in the transmission of multimedia data, such as digital images, over insecure channels, the field of securing data in transmission has gained significant attention in research. To protect data from unauthorized access, many researchers have focused on creating novel encryption methods [1], [2], and [3]. A cryptosystem is a method that consists of a group of algorithms that transform plain images into cipher images to securely encode or decode the images [4]. To assess the security strength of an encryption algorithm, it is necessary to perform a statistical analysis that typically includes measures such as entropy, correlation, energy, and homogeneity. However, the process of conducting statistical analysis frequently consumes a significant amount of time, diverting resources from the primary task at hand. Instead, we suggest that manual testing can be supplanted by a machine learning (ML) model capable of quickly, effortlessly, and accurately identifying the most robust encryption algorithm. However, transmitting data in an encrypted format is insufficient to ensure privacy. Some present image encryption techniques do not provide enough encryption of the image, allowing hackers to simply decrypt it. The resilience of an image is heavily impacted by the strength of the encryption algorithm used for its protection. A highly potent encryption will completely secure the plain image. Approach, making it resistant to attacks on its integrity, confidentiality, and availability. Beyond security, temporal complexity plays a crucial role in the choice of an encryption system. Since diverse data types come with varying security requirements, the selection of a cryptosystem should take into account the specific application's characteristics. Given the importance of image encryption methods, we offer a security-level detection approach for image encryption algorithms that incorporates an SVM. The key contributions of this research work are as follows:

1.1 Highlights

- To develop an approach for identifying security levels in image encryption algorithms using SVM.
- Compilation of necessary information from publicly available datasets.
- Development and evaluation of an SVM-driven model for analyzing the efficiency of image encryption techniques.
- Creation of a user-friendly web interface to present categorical results of encrypted images, such as Weak, Acceptable, and Strong.

The subsequent sections of this article provide a comprehensive review of the state-of-the-art literature in this re-

search domain, along with the proposed methodology, experimental setup, implementation details, results, and a concluding discussion.

2 Related work

Several encryption algorithms [5], [2], [6], [7], and [8] have been put forth in the realm of image security during transmission. However, these represent only a subset of the multitude of image encryption schemes introduced in recent years. The article [9] describes a new improved Lorenz system (ImproLS) with a larger positive Lyapunov exponent. It presents an asymmetric image encryption scheme using blind signatures and ImproLS. The experimental results demonstrate high information entropy and proximity to the theoretical value. Furthermore, it claims that both unsigned and signed cipher images can effectively withstand salt and pepper attacks as well as clipping attacks. Using the elliptic curve as the basis for image encryption [4, 5, 2] incorporates chaotic systems and ElGamal encryption. Hyperchaotic and dynamic DNA coding-based secure image encryption techniques are presented in the literature [6, 10]. Rubik's cube-based, three-dimensional bit-level image encryption scheme has been proposed by [7]. Using the upgraded Lorenz system [9] scheme, images can be encrypted effectively. A method for image encryption that is effective and is based on the upgraded Lorenz chaotic system is presented in [11]. The study presented in Arslan Shafique et al. [12] (2020) is a novel approach to image encryption employing a blending technique for securing single or multiple images of varied types and dimensions. The blending model is characterized by a non-linear mathematical expression derived from Cramer's rule. In recent times, the integration of ML models has become crucial for enhancing security and privacy in diverse applications. ML addresses critical challenges, including real-time attack detection and vulnerability assessments for data leakage. It plays a pivotal role in meeting the stringent demands of contemporary security and privacy needs, spanning areas such as real-time decision-making, big data processing, efficient learning cycles, cost-effectiveness, and error-free processing. The research work carried out by Ramani Sagar et al.[13] (2020) explores cutting-edge approaches in which ML proves to be highly effective in meeting present real-world security requirements. Authors delve into various perspectives within security applications, highlighting the indispensable role of ML models. The detection of security levels in diverse cryptosystems using an ML model is discussed in B. Manjunath et al. [14] (2020). The authors, Marcus R. Makowski et al. [15] (2020) proposed that medical imaging may become safer, private, and federated with the help of ML algorithms. Kaixin Jiao et al. [16] (2021) suggested that an image encryption algorithm can be a secure model for image communication by increasing the sensitivity of a plain image and preventing summing. The article [17] presents a hybrid cloud workflow scheduling model that enhances network security using a Lévy-optimized tSlime Mould Algorithm, demonstrating improved performance in resource allocation and threat mitigation. The study presented in [18] explores the integration of novel image processing-based feature extraction methods with artificial neural networks, resulting in more accurate multimedia content analysis. The work [19] introduces a forgery detection framework that combines Particle Swarm Optimization (PSO) and SVM, achieving high accuracy in identifying tampered electronic image data, thereby strengthening digital forensics capabilities. Researchers interested in this technique should check [8] for the implementation of the web interface; we used the technology Django[20, 21]. Table 1 presents a summary of existing research on image encryption and ML-based security techniques. Each entry outlines the author(s), publication year, and the specific feature or contribution of their work in enhancing image encryption or data protection. Various methods are highlighted, including the use of Lorenz and upgraded Lorenz chaotic systems, ML-based security assessment, and federated learning for secure medical imaging.

Table 1: Summary of related work in image encryption and machine learning-based security

Author(Year)	Features
Zou et al. [9] (2020)	Improved Lorenz system (ImproLS) with higher Lyapunov exponent
Kaur et al. [11] (2018)	Used the upgraded Lorenz chaotic system for efficient image encryption.
Arslan Shafique et al. [12] (2020)	Developed an image encryption technique using a blending model derived from Cramer's rule.
Ramani Sagar et al.[13] (2020)	Real-time attack detection, data leakage prevention, and vulnerability assessment.
Manjunath et al.[14] (2020)	To assess security levels in crypto systems using ML
Makowski et al. [15] (2020)	Suggested federated learning and privacy-preserving ML for secure medical imaging.
Kaixin Jiao et al. [16] (2021)	Increase image sensitivity and prevent data summation.

Upon thorough examination and investigation of the existing literature, we have identified opportunities for further development in this field. Our review of the literature also introduced us to the concept of SVM. Cryptography, a crucial component of computer security, plays a key role in protecting sensitive data from unauthorized access and ensuring the confidentiality, integrity, and authenticity of information. Throughout computer networking and digital communication, cryptographic methods have undergone significant advancements.

3 Dataset

To implement the ML-based application, a dataset is used to train and test ML models. For the experimental work, a dataset sourced from [12] is used. Some samples of this data set are presented in the following Table 3, which displays the security parameters, which serve as extracted features from encrypted images. These features, including measures such as entropy, energy, contrast, and others, are used to train and evaluate the performance of the ML models. Measures of an image's complexity and unpredictability, known as energy and entropy, are crucial aspects to take into account while analyzing cryptosystems. Measures of an image texture, such as correlation and homogeneity, can be used to distinguish between various cryptosystems. The PSNR and the MSE are two often-used metrics in image processing that reveal how well the encrypted image compares to the original in terms of quality. Data set labels are classified into three levels of security: Strong, Acceptable, and Weak.

Table 2: Datasets used for training and testing the machine learning models

Source	Features
Dataset 1 [22]	7
Dataset 2 [1]	7
Dataset 3 (Dataset 1 + Dataset 2)	7

Table 2 presents the sources of the datasets used in this experimental study, along with the number of key features in each dataset. Datasets 1 [1] and 2 [12] are publicly available, while dataset 3 was created to enhance the strength of the algorithm by providing additional data during the training of the ML model. As a consequence, this dataset is often referred to as the source domain for the training, validation, and construction of the proposed model. The feature column with a value of 7 indicates that seven features, entropy, energy, contrast, correlation, homogeneity, MSE, and PSNR, have been extracted from the encrypted image to assess its security level.

The Table 3 presents a detailed evaluation of 45 cipher images based on several image quality and security metrics, including Entropy, Energy, Contrast, Correlation, Homogeneity, MSE, PSNR, and Security Level. Cipher images are categorized into three security levels: Strong, Acceptable, and Weak, based on the combination of statistical characteristics and PSNR values. Generally, images with higher entropy, lower MSE, and lower PSNR fall under the "Strong" category, indicating higher encryption strength. As the PSNR increases and entropy decreases slightly, the security classification transitions from Strong to Acceptable and then to Weak.

3.1 Features of the encrypted image

Security parameters, including entropy, energy, contrast, correlation, homogeneity, histogram uniformity, and irregular deviation, serve as features for determining the level of information in plain text images.

3.1.1 Contrast

Contrast analysis reveals discrepancies in pixel values. Increased contrast indicates enhanced security, whereas lower contrast values indicate minimal distinctions between the original and altered pixel values. Mathematically, the contrast can be represented as shown in Equation 1,

$$Contrast = \sum |x - y|^2 z(x, y) \tag{1}$$

3.1.2 Entropy

The level of unpredictability of an encryption algorithm in the cipher picture is revealed through entropy analysis. Depending on the number of bits in the image, different images have varied maximum entropy values. Mathematically, the entropy can be represented as shown in Equation 2,

$$Entropy = \sum_{d=1}^{M} p(s_m) \log_2(p(s_m)) \tag{2}$$

The entropy feature handles randomness in the image more effectively.

3.1.3 Energy

This property is used to determine how much information an image contains. More information is included in the image when the energy values are higher. Equation 3 is the mathematical representation of it.,

$$Energy = \sum_{K=1}^{L} im(x, y)^2 \tag{3}$$

3.1.4 Correlation

The term "correlation" (μ_{ab}) 4 describes how closely related the pixel values are to each other. A high correlation value indicates that the pixel values are closely related. Mathematically, the correlation can be represented as an Equation 4.

$$\mu_{ab} = \frac{E[a - E(a)][y - E(b)]}{\sqrt{D(a)}\sqrt{D(b)}} \tag{4}$$

3.1.5 Homogeneity

The Gray Level Occurrence Matrix (GLCM) presents pixel brightness in a tabular format. In the context of robust encryption, it is desirable to minimize homogeneity values 5.

$$\sum_a \sum_b \frac{P(a, b)}{1 + |a - b|} \tag{5}$$

Table 3: Sample encrypted image feature's dataset [14]

Encrypted image	Entropy	Energy	Contrast	Correlation	Homogeneity	MSE	PSNR	Security level
0	8	0.01	10.75	-0.5	0.39	222	0.1	Strong
1	7.99	0.01	10.74	-0.49	0.39	221	0.2	Strong
2	7.99	0.01	10.74	-0.49	0.39	220	0.3	Strong
3	7.99	0.01	10.73	-0.48	0.39	219	0.4	Strong
4	7.99	0.01	10.73	-0.48	0.39	218	0.5	Strong
5	7.99	0.01	10.72	-0.47	0.39	217	0.6	Strong
6	7.99	0.01	10.72	-0.47	0.39	216	0.7	Strong
7	7.99	0.01	10.71	-0.46	0.39	215	0.8	Strong
8	7.99	0.01	10.71	-0.46	0.39	214	0.9	Strong
9	7.99	0.010	10.70	-0.45	0.39	213	1	Strong
10	7.99	0.01	10.7	-0.45	0.39	212	1.1	Strong
20	7.99	0.01	10.24	0.00	0.40	121	10.2	Acceptable
21	7.98	0.01	10.24	0.00	0.40	120	10.3	Acceptable
22	7.98	0.01	10.23	0.00	0.40	119	10.4	Acceptable
23	7.98	0.01	10.23	0.00	0.40	118	10.5	Acceptable
24	7.98	0.01	10.22	0.00	0.40	117	10.6	Acceptable
25	7.98	0.01	10.22	0.00	0.40	116	10.7	Acceptable
26	7.98	0.01	10.21	0.00	0.40	115	10.8	Acceptable
27	7.98	0.01	10.21	0.00	0.40	114	10.9	Acceptable
28	7.98	0.01	10.20	0.00	0.40	113	11	Acceptable
29	7.98	0.01	10.2	0.00	0.40	112	11.1	Acceptable
30	7.98	0.01	10.19	0.00	0.40	111	11.2	Acceptable
31	7.98	0.01	10.19	0.00	0.40	110	11.3	Acceptable
32	7.98	0.01	10.18	0.00	0.40	109	11.4	Acceptable
33	7.98	0.01	10.18	0.00	0.40	108	11.5	Acceptable
34	7.98	0.01	10.17	0.00	0.40	107	11.6	Acceptable
40	7.97	0.20	9.74	0.00	0.41	20	20.3	Weak
41	7.97	0.02	9.73	0.00	0.41	19	20.4	Weak
42	7.97	0.02	9.73	0.00	0.41	18	20.5	Weak
43	7.97	0.02	9.72	0.00	0.41	17	20.6	Weak
44	7.97	0.02	9.72	0.00	0.41	16	20.7	Weak
45	7.97	0.02	9.71	0.00	0.41	15	20.8	Weak

3.1.6 PSNR and MSE

PSNR is a useful feature for image encryption because it provides a quantifiable measure of the amount of distortion or scrambling introduced by the encryption process. PSNR value can be calculated between two images. Before calculating the PSNR value 6, it is necessary to calculate the MSE value 7 between the two desired images. If the PSNR value between the two images (original and cipher) is high, this means that the processed image is very close to the original image.

$$PSNR = 20\log_{10}\left(\frac{max_{val}}{\sqrt{MSE}}\right) \tag{6}$$

$$MSE = \frac{1}{XY}\sum_{a=1}^{X}\sum_{b=1}^{Y}(P_{im}(a,b) - C_{im}(a,b)) \tag{7}$$

3.1.7 Feature normalization before SVM training

Feature normalization is an essential step, as the SVM classifier relies on distance-based calculations during training. All extracted features like entropy and PSNR, are normalized to ensure they contribute equally to the learning process. We apply "StandardScaler" normalization to rescale each feature in the range [0, 1]. This normalization ensures that no single feature disproportionately influences the SVM's decision boundary due to its scale.

4 Proposed system

Over the past few years, a multitude of encryption methods have been introduced, including chaos-based and transformation-based algorithms. This research presents an ML-based (SVM) model to predict the encryption levels of the image.

4.1 Prediction of image encryption levels

Recent years have witnessed an influx of encryption algorithms, encompassing chaos and transformation-based techniques. Analyzing encryption statistics and evaluating the security configurations of these algorithms is one way to gauge their security levels. While our literature review revealed numerous image encryption methods, selecting the most suitable one for a specific purpose remains a challenging task.

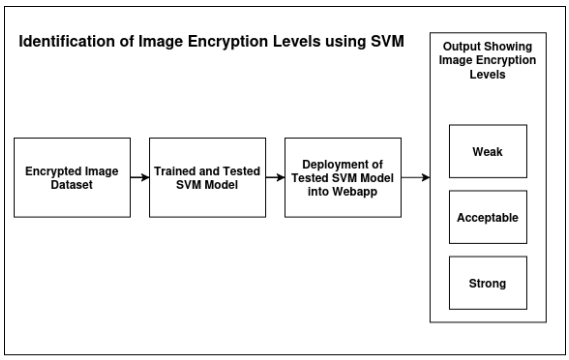

Figure 1: Block diagram of the proposed system for identifying image encryption levels – strong, weak, and acceptable – using machine learning models.

The proposed system is a web-based interface that enables users to assess the strengths of image encryption algorithms by applying ML, as shown in Figure 1. The system is designed to be user-friendly and time-saving, making it easier for users to perform analysis on their encrypted images. The user can choose an encryption scheme and upload an encrypted image to a website to use the system. After a successful analysis, the system categorizes the results as Strong, Acceptable, or Weak, based on the features extracted from the encrypted image. Overall, the proposed system is a useful tool for individuals and organizations looking to assess the strength of their encryption algorithms. By providing a user-friendly and time-saving interface, the system makes it easier for users to perform analysis on their encrypted images and make informed decisions about their encryption protocols.

4.2 Working of the proposed classifier

The goal of this ML model is to classify encrypted images into one of three categories: Weak, Acceptable, or Strong. The model uses an SVM classifier, which is a popular algorithm for classification tasks. The flow diagram of the logical data for the illustration of the working of the proposed classifier is shown in Figure 2. The input to the model is an encrypted image, which is first preprocessed to extract features such as energy, entropy, correlation, homogeneity, PSNR, and MSE. These features are then used as input to the SVM classifier. The SVM classifier is first trained on a dataset of encrypted images with known categories (i.e., Weak, Acceptable, or Strong encryption). During training,

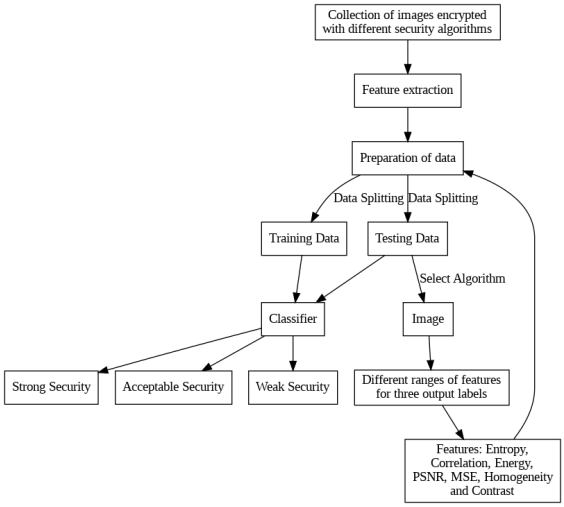

Figure 2: Logical data flow diagram illustrating the working process of the proposed classifier

the SVM model learns to recognize patterns in the input features that correspond to each category. The SVM algorithm finds the hyperplane in the high-dimensional feature space that best separates the different categories of encrypted images. Once the SVM model has been trained, it can be used to classify new encrypted images into one of the three categories: Weak, Acceptable, or Strong. The input image is first preprocessed to extract the same set of features that were used during training, and these features are then input to the SVM classifier. The SVM algorithm then maps the input image to a point in the feature space and determines which side of the decision boundary the point falls on. The category of the input image is then determined based on the position of the point relative to the decision boundary. The output of the model is a category label for the input image, which can be visualized using a scatter plot or a confusion matrix. The categories are Weak, Acceptable, and Strong correspond to different levels of encryption strength, with strong encryption being the most secure [23].

4.3 Image encryption algorithms

Image encryption algorithms are designed to protect the confidentiality of digital images by converting them into a form that is unreadable or difficult to decipher without the appropriate decryption key. DNA encoding, Logistic Map, Rubik's cube image encryption, and Lorenz image encryption are some of the commonly used image encryption algorithms, each with its own strengths and weaknesses.

4.3.1 DNA encoding

An interdisciplinary field that is rapidly expanding is DNA computing. Many biological and algebraic operations based on DNA sequences have been proposed by researchers [3]. The binary system is used to express all information in the contemporary theory of the electronic

computer. Yet, DNA sequences rather than binary numbers are used to represent information in the DNA coding hypothesis[22].

4.3.2 Logistic map

A logistic map is a degree-two polynomial mapping (also known as a recurrence relation). It is frequently used as a classic illustration of how simple non-linear equations may produce complicated, chaotic behavior. The two effects that this nonlinear difference equation is meant to convey are Confusion and Diffusion [16].

4.3.3 Rubik's cube image encryption

For real-time Internet encryption and transmission applications, a rapid encryption/decryption technique has been developed. Moreover, it has quick encryption and decryption capabilities, making it appropriate for real-time Internet encryption and transmission applications [24].

4.3.4 Lorenz image encryption

In a 3D dynamical system, the Lorenz chaotic equation is defined by x, y, and z. It suggests that the "butterfly effect" is caused by one of the traditional chaotic systems. E.N. Lorenz documented and published the model for the first time in 1963. Regarding the initial system parameters, the Lorenz chaotic equation system exhibits chaotic behavior. The Lorenz system exhibits chaotic behavior that is far more complex than any 1D or 2D chaotic system [25].

4.4 Experimental setup

All experiments and model training were performed using a cloud-based platform like Google Colaboratory (Colab), which provides free access to GPU/TPU online resources. These platforms facilitate the efficient development, testing, and deployment of ML models in a collaborative environment. The default CPU configuration for Colab is an Intel Xeon with 2 vCPUs (virtual CPUs) and a RAM size of 13 GB. The Scikit-learn library, version 1.6.1, was utilized for implementing the SVM with the RBF kernel. All extracted features were normalized using "StandardScaler" before training the SVM model. This normalization ensures that all features contribute equally to the learning process and prevents any bias due to differing value ranges.

5 Implementation

5.1 Kernel selection

The RBF kernel can map input features into an infinite-dimensional space without requiring prior knowledge about the data's structure. Unlike the polynomial kernel, which may lead to overfitting with higher degrees or underfitting with lower ones, the RBF kernel offers a more flexible and robust decision boundary with fewer parameters. The sigmoid kernel, although inspired by neural networks, often suffers from convergence issues and fails to satisfy Mercer's condition for all parameter settings, thereby limiting its applicability. The execution of this project involves several critical phases: preparing the dataset, developing an SVM classifier-based model, creating a web interface, performing analysis and presenting results, performing evaluation and fine-tuning, and finally deploying the system. During development, the system was initially divided into two primary components.

5.2 Development of model

In this section, we present the most commonly adopted steps for developing a ML model for any application. These steps include data collection and preparation, model selection, training, evaluation, parameter tuning, and making predictions. Gather a dataset comprising encrypted images generated using various encryption algorithms, key sizes, and plaintexts. Apply the selected cryptosystem algorithms to the plaintext images, resulting in distinct datasets of encrypted images for each algorithm. Extract features from these encrypted images, including measures like energy, entropy, correlation, homogeneity, PSNR, and MSE. Construct an ML model employing the SVM classifier. Train this model using the features extracted from the encrypted images as input and their respective algorithm labels as output.

5.3 Integrating a machine learning model into a web application

The interface should allow users to upload an encrypted image and select the encryption algorithm used to encrypt the image. The interface should also include a button to initiate the analysis and display the result. The implementation of a Django project involves defining models, views, and URLs, configuring settings, and testing the app. Once the user uploads an image and selects an algorithm, the system should extract the features from the image and use the SVM classifier to predict the algorithm used to encrypt the image. The system should then categorize the result as Weak, Acceptable, or Strong based on the Accuracy of the prediction. In recent times, there has been a proliferation of encryption algorithms, encompassing chaos and transformation-based approaches. Assessing encryption statistics and the security parameters of these algorithms represents a common approach for evaluating their level of security. A presented a web-based interface as part of our research article, allowing users to assess the robustness of various cryptosystems through ML model (see Figure 6.). The interface is designed to be user-friendly and time-efficient, streamlining the process of analyzing encrypted images. Users can select an encryption scheme and upload an encrypted image to the website. Leveraging a ML model developed based on current literature, the system conducts an in-depth analysis of

the uploaded encrypted image. Upon successful analysis, the system categorizes the results as Strong, Acceptable, or Weak, utilizing features extracted from the encrypted image. The user is then provided with the categorized outcomes. In summary, our proposed system offers a valuable tool for individuals and organizations seeking to evaluate the efficacy of their encryption algorithms. With its user-friendly and time-saving interface, the system facilitates a thorough analysis of encrypted images, empowering users to make informed decisions regarding their encryption protocols.

6 Performance parameters

The main performance parameters used are the confusion matrix, precision, precision, recall, and the F1 score. To define these metrics mathematically, the terminologies used are TP, FP, TN, and FN. TP and FP stand for true positive and false positive, while TN and FN represent true negative and false negative, respectively.

6.1 Confusion matrix

This is a performance metric used in ML classification tasks with two or more possible classes. It comprises a table presenting four distinct combinations of predicted and actual values.

6.2 Accuracy

Accuracy serves as a metric for assessing classification models. More formally, Accuracy is defined as follows:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{8}$$

6.3 Precision

The ratio of true positive anticipated observations to all positive predicted observations is known as Precision. This can be written mathematically as:

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

6.4 Recall

The sensitivity of the model is referred to as Recall. The model will be more sensitive to a higher Recall score. This quantifies the ratio of genuine positive observations to the number of real positive and fake negative observations combined. Recall can be calculated mathematically as:

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

6.5 F1 Score

In assessing the performance of ML models, both Accuracy and F1 score play vital roles. Accuracy is essential when giving equal importance to true positive and true negative samples, while the F1 score becomes significant when there is a greater concern for false positive and false negative samples, and vice versa. F1 score is determined by:

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{11}$$

7 Results and discussion

We utilized publicly accessible datasets, namely, Dataset 1 [12] and Dataset 2 [1], in our experimental work. Additionally, we generated Dataset 3 by randomly amalgamating samples from both Dataset 1 and Dataset 2.

Our experimentation involved the implementation and evaluation of various SVM kernels, including linear, polynomial, sigmoid, and RBF, on these datasets to assess their performance. We then employed an SVM with the radial basis function (RBF) as the kernel for training. The performance of the SVM model is evaluated using metrics such as Accuracy, Precision, Recall, and F1 score. These metrics can help determine the effectiveness of the SVM model in classifying encrypted messages and identifying potential security breaches.

7.1 Performance evaluation of machine learning model

The assessment of the developed model involves the use of three different data sets. In the upcoming sections, we will present these evaluations separately. To begin, we evaluated the model's performance using dataset 1. We divided this dataset into training and testing subsets with a 70:30 ratio and set the random state to 5. Our observations are presented in Table **??**, the following metrics for the respective categories:

Table 4: Assessing the performance of an SVM classifier with an RBF kernel using dataset 1

Performance Parameters	SVM Kernels RBF		
	Strong	Acceptable	Weak
Precision	100	88	100
Recall	100	100	83
F1 Score	100	93	91
Accuracy	98		

- For the "Acceptable" category, the developed model achieved a Precision of 88%.
- For the "strong" category, it exhibited a Precision of 100%.

– In the ”weak” category, the model also demonstrated a Precision of 100%.

The performance of the model is also presented using the confusion matrix as shown in Figure **??** This model is evaluated on the second test dataset (Dataset 2), which has been divided into training and testing subsets with a 60:40 ratio. A random state value of 5 is used to ensure reproducibility. The results presented in Table 5 indicate that the model performs exceptionally well, achieving an Accuracy rate of 97%, Precision of 100%, Recall of 88%, and an F1 score of 93% when categorizing input images encrypted with strong encryption. The ML model we've implemented, utilizing SVM, is applied to classify the image encryption level as either Strong, Acceptable, or Weak. This model is evaluated on the third test dataset (Dataset 3), which has been divided into training and testing subsets with a 60:40 ratio. A random state value of 15 is used to ensure reproducibility. We evaluated the performance of the model using various metrics, including Accuracy, Precision, Recall, and the F1 score. The Table 6 shows results indicate that the model performs exceptionally well, achieving an Accuracy rate of 98%, Precision of 100%, Recall of 94%, and an F1 score of 97% when categorizing input images encrypted with strong encryption. The confusion matrix (Figure 3)illustrates the performance of the SVM classifier in predicting encryption strength levels: ”Strong,” ”Acceptable,” and ”Weak.” It shows high classification Accuracy with 9 correct predictions for ”Strong,” 8 for ”Acceptable,” and 6 for ”Weak,” with only one misclassification where a ”Weak” sample was incorrectly predicted as ”Strong. Similarly, Figures 4 and 5 illustrate the model's performance on Dataset 2 and Dataset 3, respectively, using confusion matrices as the evaluation metric.

Table 5: Assessing the performance of an SVM classifier with an RBF kernel using dataset 2

Performance Parameters	SVM Kernels RBF		
	Strong	Acceptable	Weak
Precision	100	91	100
Recall	88	100	100
F1 Score	93	95	100
Accuracy	97		

Table 6: Assessing the performance of an SVM classifier with an RBF kernel using dataset 3

Performance Parameters	SVM Kernels RBF		
	Strong	Acceptable	Weak
Precision	100	91	100
Recall	88	100	100
F1 Score	93	95	100
Accuracy	98		

Figure 3: Confusion matrix obtained using dataset 1

Figure 4: Confusion matrix obtained using dataset 2

7.2 Performance evaluation of the web application

We have integrated the SVM model into a user-friendly web application, developed using the Django technology stack, to classify image encryption levels as Strong, Acceptable, or Weak. The system produces a strong result when we choose the DNA encryption image encryption algorithm, as shown in Figure 6. The system displays the result as Acceptable in Figure 7 if we choose the Logistic Map image encryption algorithm, and for the weak encryption algorithm, the result is weak. The final product is a user-friendly web-based tool that can automatically analyze encrypted images and categorize the encryption algorithm used as Weak, Acceptable, or Strong based on the Accuracy of the prediction.

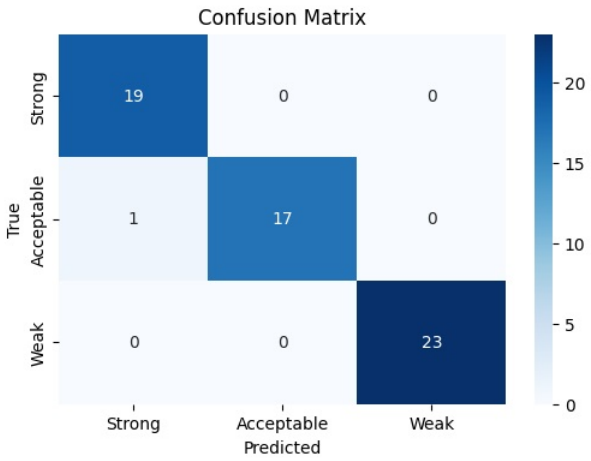

Figure 5: Confusion matrix obtained using dataset 3

Figure 6: Predicted categorical outcome: strong

7.3 Performance analysis

We have compared our proposed system's performance with the various already-published works of authors [26], [27], [28], and [29] by considering performance parameters like Accuracy, Precision, Recall, and F1 score. This comparative analysis is shown in Table 7.

Table 7: Comparative performance analysis

Methods/ Performance Parameters	Accuracy	Precision	Recall	F1 Score
Proposed System	98	100	94	97
[26]	82	96	90	92
[27]	92	93	91	83
[28]	77	90	93	81
[29]	89	87	92	92

Table 7 provides a comparative performance evaluation between the proposed system and four existing methods ([23] to [26]) using metrics such as Accuracy, Precision, Recall, and F1 Score. The proposed system demonstrates superior performance, achieving the highest values: 98% Accuracy, 100% Precision, 94% Recall, and 97% F1 Score. This enhanced performance can be attributed to the use of a larger training dataset in the proposed system compared to those used in the references.

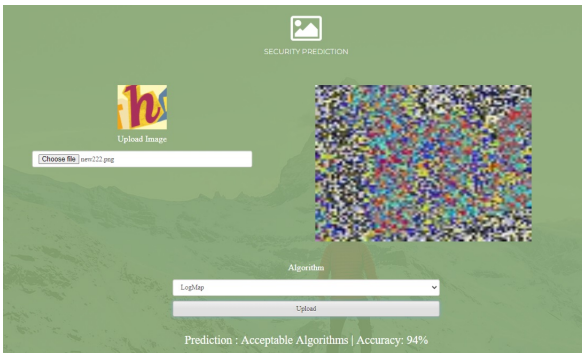

Figure 7: The categorical outcome is acceptable

7.4 Potential challenges and limitations

Key issues include latency due to network dependency, computational limitations on mobile devices for running heavier models, and ensuring responsive design for a consistent user experience.

8 Conclusion and future scope

This research proposes an SVM-based classification framework for predicting the strength of image encryption algorithms, emphasizing rapid identification and offering a user-friendly interface for efficient evaluation. Extensive experimentation with multiple kernel functions revealed that the RBF kernel achieves optimal performance in capturing the nonlinear relationships inherent in the statistical feature distributions of encrypted images. The system integrates a feature extraction pipeline with SVM modeling to classify encryption levels with high accuracy. A Django-powered web interface facilitates real-time user interaction, making the model deployable in practical security analysis contexts.

Evaluation in three benchmark data sets demonstrates the superiority of the model over existing methods, achieving accuracy up to 98%, and precision of 100% for strong encryption identification. Compared to traditional classifiers, the proposed approach offers improved discrimination capacity and operational efficiency.

The present work demonstrates the effectiveness of SVM in selecting suitable encryption techniques based on the security requirements of digital images. One promising direction is the integration of deep learning models, particularly Convolutional Neural Networks (CNNs). The research can be extended by utilizing large-scale and diverse datasets, such as domain-specific datasets (e.g., medical images) to test the robustness and generalizability of the proposed system under various security scenarios.

Code availability

The source code developed and used in this research is publicly available for reproducibility and further research. It

can be accessed at the following GitHub repository: Code

Acknowledgments

The authors acknowledge the use of AI-assisted language editing tools to enhance the clarity and readability of the manuscript. These tools were used solely for linguistic improvement and did not affect the scientific content or conclusions.

References

[1] Moatsum Alawida Abdul Nasir Khan Arslan Shafique, Abid Mehmood and Atta Ur Rehman Khan. "A novel machine learning technique for selecting suitable image encryption algorithms for iot applications". *Wireless Communications and Mobile Computing*, 2022:1–21, 2022. https://doi.org/10.1155/2022/5108331.

[2] Lu Xu, Zhi Li, Jian Li, and Wei Hua. "A novel bit-level image encryption algorithm based on chaotic maps". *Optics and Lasers in Engineering*, 78:17–25, 2016. https://doi.org/10.1016/j.optlaseng.2015.09.007.

[3] MA Ben Farah, R Guesmi, A Kachouri, and M Samet. "A novel chaos based optical image encryption using fractional fourier transform and dna sequence operation". *Optics & Laser Technology*, 121:105777, 2020. https://doi.org/10.1016/j.optlastec.2019.105777.

[4] Yuling Luo, Xue Ouyang, Junxiu Liu, and Lvchen Cao. "An image encryption method based on elliptic curve elgamal encryption and chaotic systems". *IEEE Access*, 7:38507–38522, 2019. https://doi.org/10.1109/access.2019.2906052.

[5] Shuqin Zhu and Congxu Zhu. "Plaintext-related image encryption algorithm based on block structure and five-dimensional chaotic map". *IEEE Access*, 7:147106–147118, 2019. https://doi.org/10.1109/access.2019.2946208.

[6] Shuqin Zhu and Congxu Zhu. "Secure image encryption algorithm based on hyperchaos and dynamic dna coding". *Entropy*, 22(7):772, 2020. https://doi.org/10.3390/e22070772.

[7] Hegui Zhu, Lewen Dai, Yating Liu, and Lijun Wu. "A three dimensional bit level image encryption algorithm with rubik's cube method". *Mathematics and Computers in Simulation*, 185:754–770, 2021. https://doi.org/10.1016/j.matcom.2021.02.009.

[8] Mohammed Benabdellah, Fakhita Regragui, Nourddine Zahid, and El Houssine Bouyakhf. "Encryption-compression of echographic images using fmt transform and des algorithm". *INFOCOMP Journal of Computer Science*, 6(4):36–42, Dec. 2007. https://infocomp.dcc.ufla.br/index.php/infocomp/article/view/193.

[9] Chengye Zou, Qiang Zhang, Xiaopeng Wei, and Chanjuan Liu. "Image encryption based on improved lorenz system". *IEEE Access*, 8:75728–75740, 2020. https://doi.org/10.1109/access.2020.2988880.

[10] Minal Govind Avasare and Vishakha Vivek Kelkar. "Image encryption using chaos theory". In *2015 International Conference on Communication, Information & Computing Technology (ICCICT)*, pages 1–6. IEEE, 2015. https://doi.org/10.1109/iccict.2015.7045687.

[11] M Kaur and VJEL Kumar. "Efficient image encryption method based on improved lorenz chaotic system". *Electronics Letters*, 54(9):562–564, 2018. https://doi.org/10.1049/el.2017.4426.

[12] Arslan Shafique, Jameel Ahmed, Wadii Boulila, Hamzah Ghandorh, Jawad Ahmad, and Mujeeb Ur Rehman. "Detecting the security level of various cryptosystems using machine learning models". *IEEE Access*, 9:9383–9393, 2020. https://doi.org/10.1109/access.2020.3046528.

[13] Ramani Sagar, Rutvij Jhaveri, and Carlos Borrego. "Applications in security and evasions in machine learning: a survey". *Electronics*, 9(1):97, 2020. https://doi.org/10.3390/electronics9010097.

[14] M Gomathi, S Gunasekar, S Abinesh, et al. "Detecting the security level of various cryptosystems using machine learning model". *International Journal of Progressive Research in Science and Engineering*, 3(05):25–31, 2022. https://journal.ijprse.com/index.php/ijprse/article/view/557.

[15] Georgios A Kaissis, Marcus R Makowski, Daniel Rückert, and Rickmer F Braren. "Secure, privacy-preserving and federated machine learning in medical imaging". *Nature Machine Intelligence*, 2(6):305–311, 2020. https://doi.org/10.1038/s42256-020-0186-1.

[16] Jakub Oravec, Lubos Ovsenik, and Jan Papaj. "An image encryption algorithm using logistic map with plaintext-related parameter values. *Entropy*, 23(11):1373, 2021. https://doi.org/10.3390/e23111373.

[17] Jingang Guan. "Enhanced network security hybrid cloud workflow scheduling using levy-optimized

slime mould algorithm". *Informatica An International Journal of Computing and Informatics*, 49:111–126, 2025. https://doi.org/10.31449/inf.v49i18.7327.

[18] Hong shun Chen Lianqiu Liu1, Yongping Yang. "Application of new feature techniques for multimedia analysis in artificial neural networks by using image processing". *Informatica An International Journal of Computing and Informatics*, 48:113–124, 2024. https://doi.org/10.31449/inf.v48i11.5851.

[19] Yang Lei Lingyu Liao. "Image content forgery detection model combining pso and svm in electronic data forensics". *Informatica An International Journal of Computing and Informatics*, 48:113–124, 2024. https://doi.org/10.31449/inf.v48i8.5897.

[20] William S. Vincent. *Django for Beginners: Build websites with Python and Django*. WelcomeToCode, 1st edition, 2018. DOI not available.

[21] Daniel Rubio. *Beginning Django Web Application Development and Deployment with Python"*. Springer, 2017. https://link.springer.com/book/10.1007/978-1-4842-2787-9.

[22] Arslan Shafique and Jameel Ahmed. "Dynamic substitution-based encryption algorithm for highly correlated data". *Multidimensional Systems and Signal Processing*, 32:91–114, 2021. https://doi.org/10.1007/s11045-020-00730-3.

[23] Peter Harrington. *"Machine Learning in Action"*. Manning Publications, Shelter Island, NY, 2012. https://dl.acm.org/doi/10.5555/2361796.

[24] Khaled Loukhaoukha, Jean-Yves Chouinard, and Abdellah Berdai. "A secure image encryption algorithm based on rubik's cube principle". *Journal of Electrical and Computer Engineering*, 2012, 2012. https://doi.org/10.1155/2012/173931.

[25] Riguang Lin and Sheng Li. "An image encryption scheme based on lorenz hyperchaotic system and rsa algorithm". *Security and Communication Networks*, 2021, 2021. https://doi.org/10.1155/2021/5586959.

[26] Muhammad U Ilyas and Soltan Abed Alharbi. "Machine learning approaches to network intrusion detection for contemporary internet traffic". *Computing*, 104(5):1061–1076, 2022. https://doi.org/10.1007/s00607-021-01050-5.

[27] Sudhakar Sengan, Osamah Ibrahim Khalaf, Dilip Kumar Sharma, Abdulsattar Abdullah Hamad, et al. "Secured and privacy-based ids for healthcare systems on e-medical data using machine learning approach". *International Journal of Reliable and Quality E-Healthcare (IJRQEH)*, 11(3):1–11, 2022. https://doi.org/10.4018/ijrqeh.289175.

[28] Shashvi Mishra and Amit Kumar Tyagi. "The role of machine learning techniques in internet of things-based cloud applications". *Artificial intelligence-based internet of things systems*, pages 105–135, 2022. https://doi.org/10.1007/978-3-030-87059-1_4.

[29] Saket Acharya, Umashankar Rawat, and Roheet Bhatnagar. "A comprehensive review of android security: Threats, vulnerabilities, malware detection, and analysis". *Security and Communication Networks*, 2022, 2022. https://doi.org/10.1155/2022/7775917.