Detecting The Security Level of Various Cryptosystems Using Machine Learning Models

Submitted in partial fulfilment of the requirements for the award

of

Bachelor of Engineering degree in Computer Science and Engineering

By

Kantipudi Pranathi (Reg. No – 39110448)

Bodepudi Lakshmi Priya (Reg. No – 39110174)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHISALAI,

CHENNAI - 600119

APRIL - 2023

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Kantipudi Pranathi (Reg no- 39110448)** who carried out the Project Phase-2 entitled **"Detecting the Security Level of Various Cryptosystems Using Machine Learning Models"** under my supervision from October 2022 to April 2023.

yovanfelix

Internal Guide

Dr.A.Yovan Felix, M.E., Ph.D.,

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.,

Submitted for Viva voce Examination held on 20.4.2023

Internal Examiner

External Examiner

DECLARATION

I, **Kantipudi Pranathi (Reg no- 39110448),** hereby declare that the Project Report entitled **"Detecting the Security Level of Various Cryptosystems Using Machine Learning Models"** done by me under the guidance of **Dr. A.Yovan Felix, M.E., Ph.D** is submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

Kantipudi Pranathi

DATE: 20-04-2023

PLACE: Chennai **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph. D**, **Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.,** Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.A.Yovan Felix M.E.,Ph.D,** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The security of digital data has become a crucial concern as a result of recent advances in multimedia technology. Researchers tend to focus their efforts on changing existing protocols to overcome the flaws of present security mechanisms. Several proposed encryption algorithms, however, have been proven insecure during the last few decades, posing serious security risks to sensitive data. Using the most appropriate encryption technique to protect against such assaults is critical, but which algorithm is most suited in any given case will depend on the type of data being protected. However, testing potential cryptosystems one by one to find the best option can take up an important processing time. For a fast and accurate selection of appropriate encryption algorithms, we propose a security level detection approach for image encryption algorithms by incorporating a support vector machine (SVM). In this work, we also create a dataset using standard encryption security parameters, such as entropy, contrast, homogeneity, peak signal to noise ratio, mean square error, energy, and correlation. These parameters are taken as features extracted from different cipher images. Dataset labels are divided into three categories based on their security level: strong, acceptable, and weak. To evaluate the performance of our proposed model, we have calculated accuracy and our results demonstrate the effectiveness of this SVM-supported system.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVATIONS

ABBREVIATION	EXPANSION
SVM	Support Vector Machine
XG Boost	Extreme Gradient Boost
AES	Advanced Encryption Standard
PSNR	Peak Signal to Noise Ratio
MSE	Mean Square Error
ML	Machine Learning
AI	Artificial Intelligence

CHAPTER 1

INTRODUCTION

Security has become a hot topic of research due to the exponential rise in multimedia data transmissions over insecure channels (most notably the Internet). Many researchers have turned to developing new encryption algorithms to protect data from eavesdroppers and unauthorized users. Two elements are critical when encrypting digital images: diffusion and confusion (also known as scrambling). Claud Shannon suggested a theory in which a cryptosystem with confusion and diffusion mechanisms can be considered secure. The scrambling process on digital images can be done directly on pixels or on rows and columns, whereas diffusion affects the original pixel values. In other words, the replacement procedure substitutes each unique pixel value with the S-unique box's value. The transmission of data in an encrypted format, however, is insufficient to preserve its privacy. Although the information which is to be transmitted is in encrypted form, it can still be visualized by unauthorized users due to the weak security of the encryption algorithm. The security level of the encryption algorithm used to encrypt the image has a significant impact on its robustness. The plain image will be entirely encrypted using a highly strong encryption method, allowing it to withstand attacks on its integrity, secrecy, and availability. Along with security, temporal complexity is another key element to consider when choosing an encryption technology. Because different types of data have different security priorities, choosing a cryptosystem is dependent on the nature of the application to be encrypted. As the image encryption algorithm is very important, we propose a security level detection approach for image encryption algorithms by incorporating a support vector machine (SVM).

Making strategies based on customer behaviour modification with respect to challenges generated through out time has been difficult due to frequent marker changes. This research aims to address this problem.

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. When encrypting digital images, two factors are crucial: diffusion and confusion (also known as scrambling). In, 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 original pixel values. In other words, with the substitution process, every unique pixel value replaces with the unique value of the S-box.

However, the transmission of data in an encrypted form is not enough to ensure its privacy. For instance, if anyone encrypts an image with a single substitution box (S-box), the information in the substituted or enciphered image may still be visible. This means that the encryption with a single S-box is not enough to conceal the original image properly. Although the information which is to be transmitted is in encrypted form, it can still be visualized by unauthorized users due to the weak security of the encryption algorithm. Thus. it is also necessary to use a strong encryption algorithm to boost encryption security. The robustness of the encrypted image is highly dependent on the security level of the encryption algorithm that has encrypted it. A highly secure encryption algorithm will encrypt the plain image completely, enabling it to resist attacks against its integrity, confidentiality, and availability. Along with security, time complexity is another important factor to count in the selection of an appropriate encryption system.

The selection of any cryptosystem depends on the nature of the application to be encrypted, as different types of data will have different security priorities. For example, the Advanced Encryption Standard (AES) is currently the most secure encryption algorithm available. However, it is not suitable for applications where fast encryption is required, since AES required several rounds, which takes more time to encrypt the original information. Moreover, the time complexity is also dependent on the total number of pixels present in the original image. The greater number of pixels in the plain image, the more processing time will be required to encrypt it. By contrast, if the main requirement is only to encrypt a plain image with strong security,

then the processing time may not need to such a strong consideration. Although strong encryption provides better security results, it is not necessarily a feature of fast encryption, which may be preferred sometimes. To evaluate the security level of an encryption algorithm, a statistical analysis such as entropy, correlation, energy, or homogeneity must be performed upon it. Such tasks can be achieved by testing each encryption algorithm and calculating the statistics of its security parameters.

After performing such security analyses on different encryption algorithms one by one, we can choose the best and strongest option from those tested. However, this process often takes too much time away from achieving the actual task. Instead, we propose, this manual testing can be replaced by a machine learning model, which will be able to select the strongest encryption algorithm quickly, easily, and accurately. We have categorized the security of encryption algorithms into three different levels (strong, moderate, and weak) based on standard security parameters of the encryption algorithms. Below is the detail of how we divided the encryption algorithms into three said security levels based on the security parameters such as entropy, homogeneity, contrast, correlation, energy, PSNR and MSE.

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 single Substitution-box (S-box) algorithm 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. To justify the above statement, we obtain the security parameter values for different enciphered images which are generated from different encryption algorithms. Weak

and moderate encryption algorithms are not able to encrypt the images properly. The enciphered images encrypted with weak and moderate encryption algorithms. The statistical values for different images encrypted with weak, moderate, and strong encryption algorithms and their corresponding average entropy values. For the security level detection, we have considered all types of image encryption algorithms whether it is based on the frequency domain, transform-based or chaotic maps-based schemes The main objective of the proposed work is to find the security level of the encryption algorithms.

To generate a dataset, we considered a bunch of enciphered images and extract the feature values of those images. The size of the dataset is not restricted, it can be of any size. Feature values for strong and acceptable security level must be properly mentioned in the dataset. Take entropy values as an example; for the entropy values, we have taken the step size of 0.0001. we have divided the entropy values into three said intervals. For strong security, there are one hundred values ranges from 7.9901 to 8.000. Likewise, for the acceptable security level, there are one hundred and two values range from 7.9900 to 7.9800. All the other values which are below 7.9800 will be for weak security status. Similarly, we have divided the other parameter values into three intervals by selecting an appropriate step size accordingly.

For the visualization of the dataset, some portion of the proposed dataset in which the first twenty feature vectors of each category of security level are displayed. Rules for classification: To classify the encryption algorithms into three different categories (strong, acceptable 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, atleast 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.

CHAPTER 2

LITERATURE SURVEY

Automated detection and classification of cryptographic algorithms in binary programs through machine learning by Diane Duros Hosfelt (2015)

Threats from the internet, particularly malicious software (i.e., malware) often use cryptographic algorithms to disguise their actions and even to take control of a victim's system (as in the case of ransom ware). Malware and other threats proliferate too quickly for the time-consuming traditional methods of binary analysis to be effective. By automating detection and classification of cryptographic algorithms, we can speed program analysis and more efficiently combat malware. This thesis will offer different ways for automatically discovering and classifying cryptographic algorithms in compiled binary programs using machine learning. While more research is needed to fully test these methods on real-world binary programs, the findings in this paper imply that machine learning may be used to detect and identify cryptographic primitives in compiled code with success. These techniques are now being used to discover and categorize cryptographic algorithms in small single-purpose programs, and more work is being suggested to apply them to realworld situations.

Summary

This thesis examined the process of extracting features and training machine learning models for the detection and classification of cryptographic algorithms in compiled code. Three different types of models were evaluated on four different feature sets using four different learning algorithms. While the decision tree models were found to perform the best on these data, due to certain limitations of decision trees, it is possible that an SVM with a linear kernel will generalize better to realworld data. Cross-validation results suggest that algorithm classification and detection will be over 95% accurate, given a relatively small and homogeneous sample.

Applications in Security and Evasions in Machine Learning: A Survey by Ramani Sagar 1, *, Rutvij Jhaveri 2 and Carlos Borrego 3 (2020)

In recent years, machine learning (ML) has become an important part to yield security and privacy in various applications. ML is used to address serious issues such as real-time attack detection, data leakage vulnerability assessments and many more. ML extensively supports the demanding requirements of the current scenario of security and privacy across a range of areas such as real-time decision-making, big data processing, reduced cycle time for learning, costefficiency, and error-free processing. Therefore, in this paper, we review the stateof-the-art approaches where ML is applicable more effectively to fulfill current realworld requirements in security. We examine different security applications' perspectives where ML models play an essential role and compare, with different possible dimensions, their accuracy results. By analyzing ML algorithms in security application, it provides a blueprint for an interdisciplinary research area. Even with the use of current sophisticated technology and tools, attackers can evade the ML models by committing adversarial attacks. Therefore, requirements rise to assess the vulnerability in the ML models to cope up with the adversarial attacks at the time of development. Accordingly, as a supplement to this point, we also analyze the different types of adversarial attacks on the ML models. To give proper visualization of security properties, we have represented the threat model and defense strategies against adversarial attack methods. Moreover, we illustrate the adversarial attacks based on the attackers' knowledge about the model and addressed the point of the model at which possible attacks may be committed.

Finally, we also investigate different types of properties of the adversarial attacks

Summary

New threats caused by cyber-attacks can damage critical data infrastructure because of machine learning in the security applications highly dependent on the data quality. Using machine learning-based methods in security applications faces a challenge the performance of recognizing an adversarial sample by collecting and predicting adversarial samples. Hence, we conclude that the new models are becoming a research point from attacker and designer perspective. With the rapid increase in security events security in machine learning-based decision systems in

adversarial environments opens a door for the new research area. In some cases, malicious users can simply increase false-negative rates and minimizing falsepositive rates by a proportional amount, cleverly make sure that the overall error rate remains the same and attack is unnoticed which can give attackers some leverage in sophisticated attacks. This kind of issue there needs to be explored to detect attacks efficiently on ML-based systems. Regardless of the data privacy field, great advancement in existing methods of data privacy suffer from modest performance due to complex operations on a huge number of parameters of machine learning algorithms. Therefore, extremely efficient privacy-preserving methods need to be investigated in the adversarial environmental setting. Observation made related performance tradeoff between accuracy and scalability for the machine learning classifiers. For example, for any security application designing informal decision made on which approach to use when. But even though having more weak labels does not imply that classifiers' accuracy will eventually reach a precise accuracy. Therefore, it is worth to infuse humans or utilizing transfer learning to make additional changes. This type of decision is made by an experiment, but an important question is whether, overall, there is a need to design and craft secure machine learning algorithms that way which can balance three aspects that are performance overhead, security optimization, and performance generalization

Machine learning approaches to IoT security: A systematic literature review By Rasheed Ahmada, Izzat Alsmadi (2021)

With the continuous expansion and evolution of IoT applications, attacks on those IoT applications continue to grow rapidly. In this systematic literature review (SLR) paper, our goal is to provide a research asset to researchers on recent research trends in IoT security. As the main driver of our SLR paper, we proposed six research questions related to IoT security and machine learning. This extensive literature survey on the most recent publications in IoT security identified a few key research trends that will drive future research in this field. With the rapid growth of large scale IoT attacks, it is important to develop models that can integrate state of the art techniques and technologies from big data and machine learning. Accuracy and efficiency are key quality factors in finding the best algorithms and models to detect IoT attacks in real or near real-time.

Summary

This paper aims to investigate research trends for the applications of machine learning in IoT security. We adopted a systematic approach to evaluating recent studies and future trends in IoT security by extracting the most relevant and scholarly literature published in the last two years (2019 and 2020). Our objectives are driven by the interest to integrate three trending research areas, IoT, machine learning, and information security. The integration of those three domains leads us to extract six research questions presented in Section 3.1. The extensive but systematic literature review approach presented in this paper shows specific criteria used to filter research papers that did not meet this research goal. It helped us obtain more focused and recent research studies in IoT and the machine learning techniques proposed to prevent large-scale attacks impacting IoT devices. While many survey studies were performed in IoT security research, they were either not performed systematically or were not focused on machine learning and deep learning-based approaches to detect large-scale attacks. Section 5 presents the details of existing IoT security literature surveys. This research study primarily focused on securing IoT devices from distributed and wide-scale attacks such as a distributed denial of service (DDoS), botnet, etc. Compare to traditional intrusion detection techniques such as firewalls, antiviruses, etc., machine learning, and deep learning approaches provide promising results to detect zero-day threats. There has been extensive research performed in this field; however, the fastevolving nature of IoT device types, network traffic patterns, and cyber-attacks make it challenging for intrusion detection systems to detect zero-day attacks. We presented a detailed background section to equip the reader with relevant knowledge to navigate further into the paper efficiently. The careful analysis of selected papers in section 6 extracts researchers proposed methodologies and their related performance results on various benchmark datasets. The review results presented in section 7 answers each of the six research questions in detail. Our goal is to provide researchers a more focused yet detailed knowledge and prevailing methodologies adopted recently to find recent trends, limitations, and challenges to help build effective intrusion detection systems in the future.

Secure, privacy-preserving, and federated machine learning in medical imaging By Georgios A. Kaissis, Marcus R. Makowski, Daniel Rückert & Rickmer F. Braren (2018)

The broad application of artificial intelligence techniques in medicine is currently hindered by limited dataset availability for algorithm training and validation, due to the absence of standardized electronic medical records, and strict legal and ethical requirements to protect patient privacy. In medical imaging, harmonized data exchange formats such as Digital Imaging and Communication in Medicine and electronic data storage are the standard, partially addressing the first issue, but the requirements for privacy preservation are equally strict. To prevent patient privacy compromise while promoting scientific research on large datasets that aims to improve patient care, the implementation of technical solutions to simultaneously address the demands for data protection and utilization is mandatory. Here we present an overview of current and next-generation methods for federated, secure and privacy-preserving artificial intelligence with a focus on medical imaging applications, alongside potential attack vectors and future prospects in medical imaging and beyond.

Summary

Artificial intelligence (AI) methods have the potential to revolutionize the domain of medicine, as witnessed, for example, in medical imaging, where the application of computer vision techniques, traditional machine learning and—more recently—deep neural networks have achieved remarkable successes. This progress can be ascribed to the release of large, curated corpora of images (Image Net perhaps being the best known), giving rise to performant pre-trained algorithms that facilitate transfer learning and led to increasing publications both in oncology—with applications in tumour detection, genomic characterization, tumour subtyping, grading prediction, outcome risk assessment or risk of relapse quantification—and non-oncologic applications, such as chest X-ray analysis and retinal fundus imaging.

Machine Learning and Cryptographic Algorithms –Analysis and Design in Ransomware and Vulnerabilities Detection by Nandkumar Niture (2020)

The AI, deep learning and machine learning algorithms are gaining the ground in every application domain of information technology including information security.

In formation security domain knows for traditional password management systems, autoprovisioning systems and user information management systems. There is another raising concern on the application and system level security with ransomware. On the existing systems cyber-attacks of Ransom ware asking for ransom increasing every day. Ransomware is the class of malware where the goal is to gain the data through encryption mechanism and render back with the ransom. The ransomware attacks are mainly on the vulnerable systems which are exposed to the network with weak security measures. With the help of machine learning algorithms, the pattern of the attacks can be analysed. Create or discuss a workaround solution of a machine learning model with combination of cryptographic algorithm which will enhance the effectiveness of the system response to the possible attacks. The other part of the problem, which is hard part to create intelligence for the organizations for preventing the ransomware attacks with the help of intelligent system password management and intelligent account provisioning. In this paper I elaborate on the machine learning algorithms analysis for the intelligent ransomware detection problem, later part of this paper would be design of the algorithm

Summary

In this paper, proposed machine learning algorithm can be modeled for the novel ransomware detection and the random number decryption techniques for the new model to break the encryption for saving the ransom. The study also suggested for the classification of the infectious files can be differentiated by the machine based on the model it has trained. The main problem has structurally divided into sub problems of the identification of the ransomware problems and the design the cryptographic algorithms based on the machine learning to generate the decryption key for the ransom problem. 8 19 April 2020 Machine Learning and Cryptographic Algorithms – Analysis and Design in Ransomware and Vulnerabilities Detection My ongoing work will be a round producing the results and accuracy of the algorithm which I am working on for my research.

Prediction of DDoS Attacksusing Machine Learning and Deep Learning Algorithms by Saritha, B. RamaSubba Reddy, A Suresh Babu (2017)

With the emergence of network-based computing technologies like Cloud

Computing, Fog Computing and IoT (Internet of Things), the context of digitizing the confidential data over the network is being adopted by various organizations where the security of that sensitive data is considered as a major concern. Over a decade there is a massive growth in the usage of internet along with the technological advancements that demand the need for the development of efficient security algorithms that could withstand various patterns of the security breaches. The DDoS attack is the most significant network-based attack in the domain of computer security that disrupts the internet traffic of the target server. This study mainly focuses to identify the advancements and research gaps in the development of efficient security algorithms addressing DDoS attacks in various ubiquitous network environments. Keywords: DDoS attack, machine learning, Deep learning, Volumetric attacks, protocol attacks.

Now a day's with the advent of 4G, 5G networks and economic smart devices there is a massive growth in the usage of the internet that has become a part of daily life. A vast range of services provided over the internet in diverse application areas such as business, entertainment, and education, etc. made it a vital component in framing various business models. This context made security over wireless networks as the most important factor while using the internet from unsecured connections [1]. Different security algorithms and frameworks are developed to enable protection from Internet-based attacks while devising high performance IDS (Intrusion detection systems) which act as a defensive wall while confronting the attacks over internet-based devices. Distributed architecture-based computing environments like cloud computing and IoT are more prone towards DDoS attacks in which multiple devices are coordinated to launch attacks over distributed targets. DDOS attacks are primarily launched in the context of exhausting the connectivity and the processing of the target server resources in which it enables the access constraints to the legitimate users to utilize the services provided by the target server that leads towards the partial unavailability or total unavailability of the services. The phenomenon of distributed computing is based on the one-tomany dimension in which these types of attacks may cause a possible amount of damage to the server resources [3]. It is observed from the previous research studies that the damage capacity, as well as the disrupting nature of the DDoS attacks, is gradually increased with the rate of internet usage.

Summary

The article includes the systematic study of literature in the context of detecting and predicting DDoS attacks on utilizing machine learning and deep learning algorithms. In this study, after the thorough filtering process, 34 articles are considered for the study through which it is observed that most of the existing research includes solutions and algorithmic patterns that are framed based on the statistical algorithms such that most these algorithms suffer from computational complexity. Additionally, there are very few publications addressing the scope of predicting the DDoS. This study outlines the synthesis of various DDoS attacks and their countermeasure algorithms that enables the researchers of the next generation to easily identify the research gap in the context of applying machine learning algorithms to automate the process of predicting DDoS attacks in various distributed networks.

Inferences from Literature Survey

- While the decision tree models were found to perform the best on these data, due to certain limitations of decision trees, it is possible that an SVM with a linear kernel will generalize better to real-world data.
- the new models are becoming a research point from attacker and designer perspective.
- This research study primarily focused on securing IoT devices from distributed and wide-scale attacks such as a distributed denial of service (DDoS), botnet.

Open problems in Existing System

Existing Method:

In the existing system, acquiring perfectly balanced and highly related dataset is almost impossible. Although large quantities of data are available but still extracting relevant data is a complex job. To overcome all this, we use machine learning packages available in the scikit-learn library to extract useful data.

Disadvantages:

- High complexity.
- Time consuming.

CHAPTER 3

REQUIREMENT ANALYSIS

FEASIBILITY STUDIES/RISK ANALYSIS OF THE PROJECT

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

Economic feasibility:

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

Technical feasibility:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

Social feasibility:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The

level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT

Operating system	: Windows 7 or 7+
Ram	: 8 GB
Hard disc or SSD	: More than 500 GB
Processor	: Intel 3rd generation or high or Ryzen with 8 GB Ram
Software's	: Python 3.6 or high version, Visual studio, PyCharm.

Functional and non-functional requirements:

Requirement's analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and non-functional requirements.

Functional Requirements: These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Examples of functional requirements:

1) Authentication of user whenever he/she logs into the system
2) System shutdown in case of a cyber-attack
3) A verification email is sent to user whenever he/she register for the first time on some software system.

Non-functional requirements: These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other.

They are also called non-behavioral requirements. They basically deal with issues like:

- Portability
- Security
- Maintainability
- Reliability
- Scalability
- Performance
- Reusability
- Flexibility

Examples of non-functional requirements:

1) Emails should be sent with a latency of no greater than 12 hours from such an activity.
2) The processing of each request should be done within 10 seconds 3) The site should load in 3 seconds whenever of simultaneous users are > 10000.

SYSTEM USECASE

1. In the Unified Modelling Language (UML), a use case diagram is a form of behavioural diagram that is defined by and produced by a use-case study.

2. Its objective is to offer a graphical picture of a system's functionality in terms of actors, their objectives (expressed as use cases), and any interdependencies between those use cases.

3. A use case diagram's primary objective is to identify which system functions are carried out for each actor. The system's actors' roles can be illustrated.

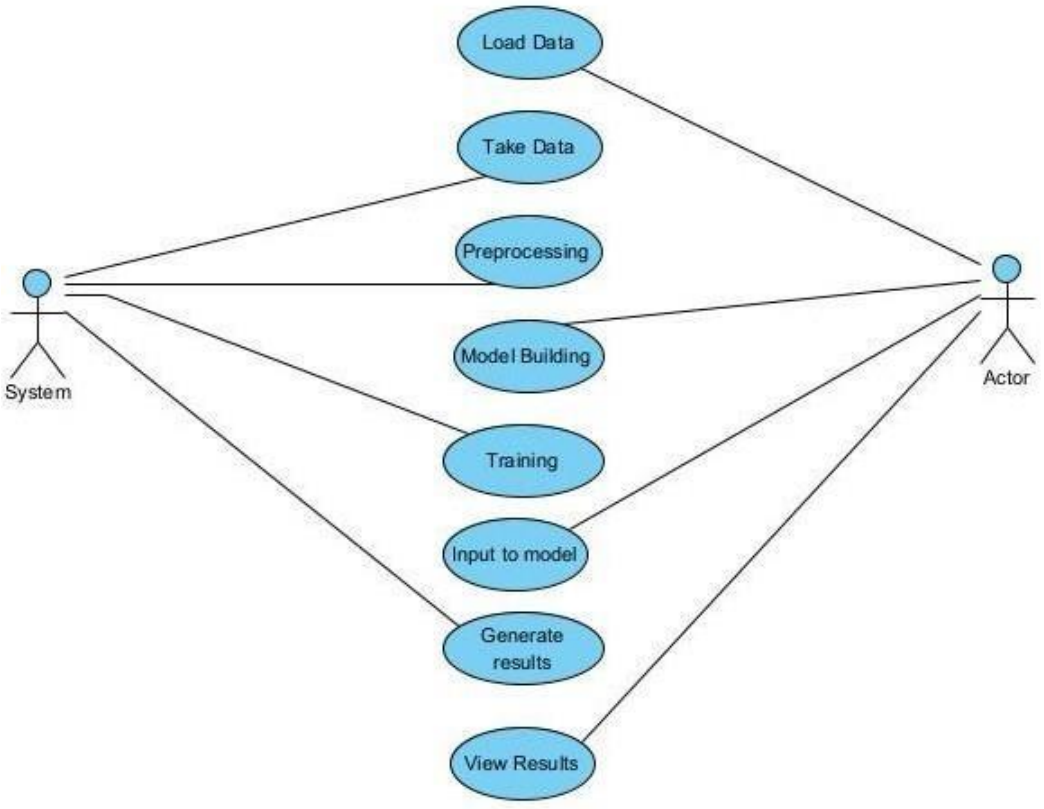


Fig 3.1: USECASE DIAGRAM

1) CLASS DIAGRAM:

A class diagram is a form of static structure diagram used in software engineering that displays the classes, attributes, operations (or methods), and interactions between the classes to illustrate the structure of a system. What class carries information is explained.

Fig 3.2: CLASS DIAGRAM

2) SEQUENCE DIAGRAM:

An interaction diagram that depicts how processes interact with one another and in what order is known as a sequence diagram in the Unified Modeling Language (UML).

It is a Message Sequence Chart construct. Event diagrams, event situations, and timing diagrams are other names for sequence diagrams.

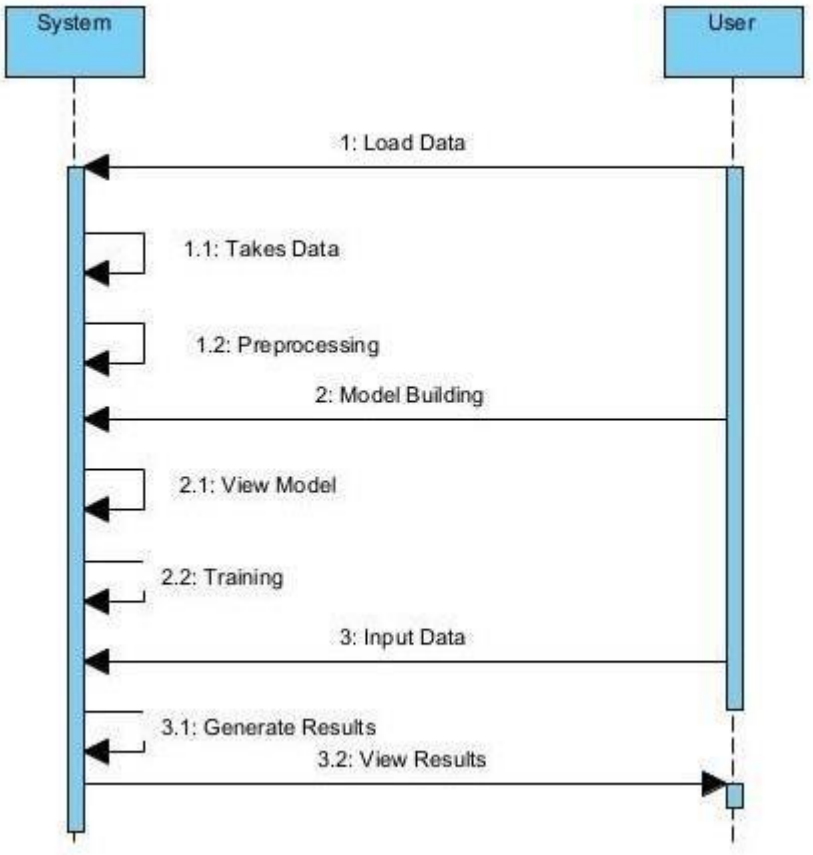


Fig 3.3: SEQUENCE DIAGRAM

3) COLLABORATION DIAGRAM:

The following cooperation diagram uses a numbering scheme to represent the order in which the methods are called. The number designates the order in which the methods are called. The cooperation diagram is described using the same order

management system. Comparable to a sequence diagram, the method calls are also similar. The cooperation diagram depicts the object organisation, but the sequence diagram does not, and this is the distinction.

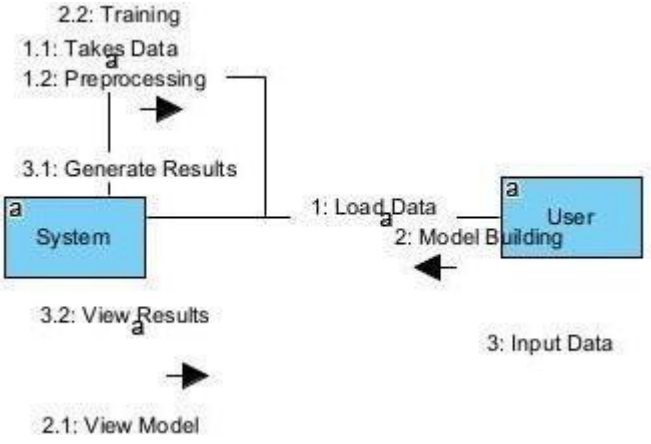


Fig 3.4: COLLBORATION DIAGRAM

4) DEPLOYMENT DIAGRAM

The deployment view of a system is represented by a deployment diagram. This and the component diagram are connected. Because deployment diagrams are used to deploy the components. In a deployment diagram, nodes are present. Nodes are only the actual pieces of hardware that are utilized to deliver the program.

Fig 3.5: DEPLOYMENT DIAGRAM

5) ACTIVITY DIAGRAM:

Activity diagrams are visual depictions of processes with choice, iteration, and concurrency supported by activities and actions. Activity diagrams may be used to depict the operational and business processes of system components in the Unified Modelling Language. An activity diagram displays the total control flow.

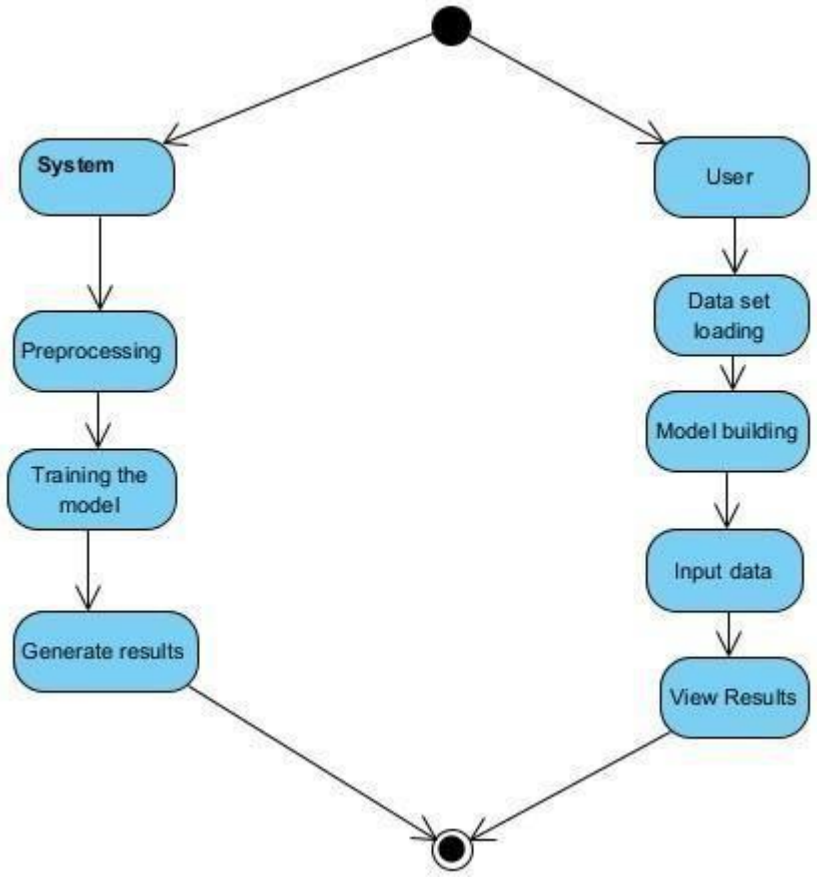


Fig 3.6: ACTIVITY DIAGRAM

6) COMPONENT DIAGRAM:

A component diagram, often called a UML component diagram, shows how the physical parts of a system are wired up and organized. To represent implementation

specifics and confirm that every part of the system's necessary function is covered by planned development, component diagrams are frequently developed.

Fig 3.7: COMPONENT DIAGRAM

7) ER DIAGRAM:

An entity-relationship model (ER model) uses an entity relationship diagram to illustrate how the structure of a database is described (ER Diagram). A database design or blueprint known as an ER model can be used to create a database in the future. The entity set and relationship set are the two fundamental parts of the E-R model.

An ER diagram illustrates the connections between entity sets. An entity set is a collection of related entities, each of which may have properties. An entity in a DBMS is a table or an attribute of a table, hence the ER diagram illustrates the whole logical structure of a database by displaying the relationships between tables and their attributes. Let's look at a straightforward ER diagram.

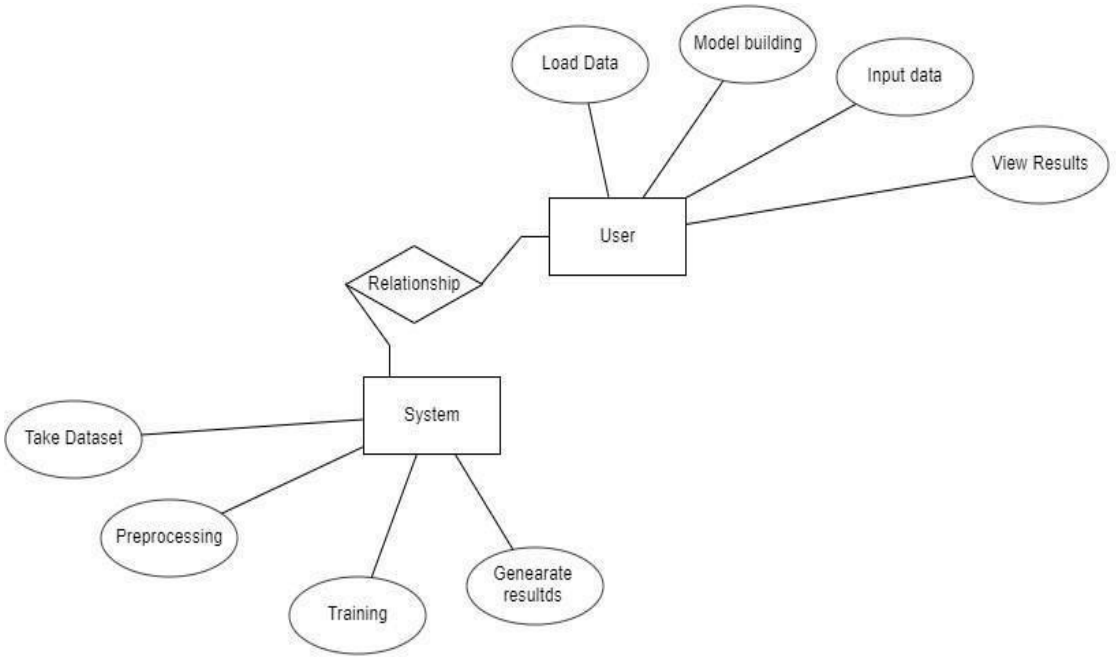


Fig 3.8: ER DIAGRAM

8) DFD DIAGRAM:

The typical method for representing the information flows inside a system is a data flow diagram (DFD). A good deal of the system requirements may be graphically represented by a tidy and understandable DFD. It can be done manually, automatically, or both. It demonstrates how data enters and exits the system, what modifies data, and where it is kept. A DFD is used to illustrate the scope and bounds of a system as a whole. It may be utilized as a medium for communication between a systems analyst and any participant in the system that serves as the foundation for system redesign.

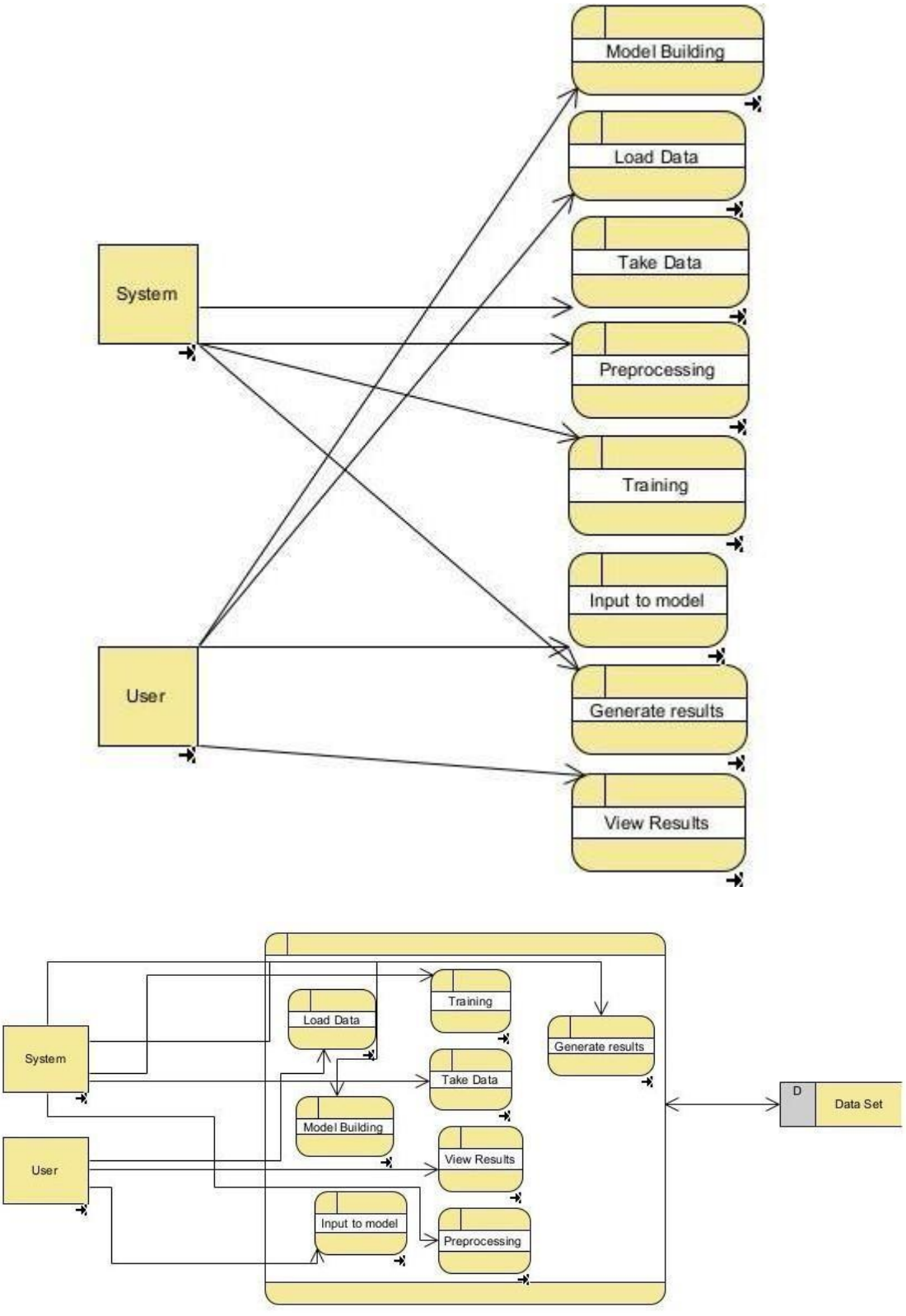


Fig 3.9: DFD DIAGRAM

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

A slew of encryption algorithms, including chaos and transformation-based algorithms, have been presented in recent years. By examining the statistical findings of existing encryption algorithms, it has been discovered that some of them are unsecure and do not provide adequate protection. Analyzing the statistics of an encryption algorithm's security parameters is one technique to determine its security level. Traditional methods of accomplishing this usually include making these comparisons one by one, which takes a long time. We created a machine learning model that combines SVM to help us choose a suitable encryption technique more rapidly.

SELECTED METHODOLOGY OR PROCESS MODEL

In our project we use waterfall model as our software development cycle because of its step-by-step procedure while implementing.

Requirement Gathering and analysis − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

System Design − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

Implementation − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

Integration and Testing − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

Deployment of system − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

Maintenance − There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

ARCHITECTURE / OVERALL DESIGN OF PROPOSED SYSTEM

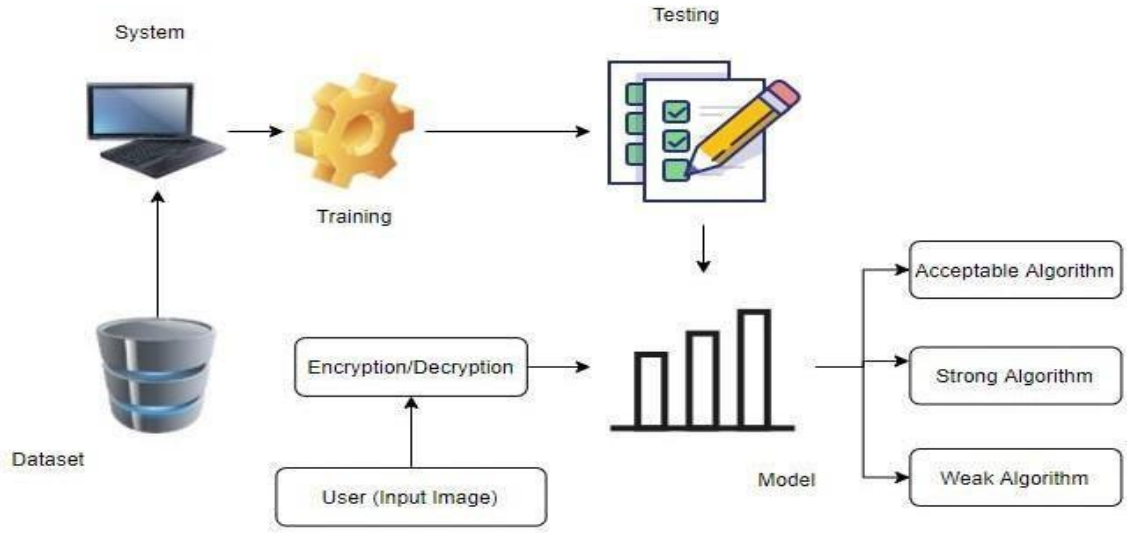


Fig 4.1: System Architecture

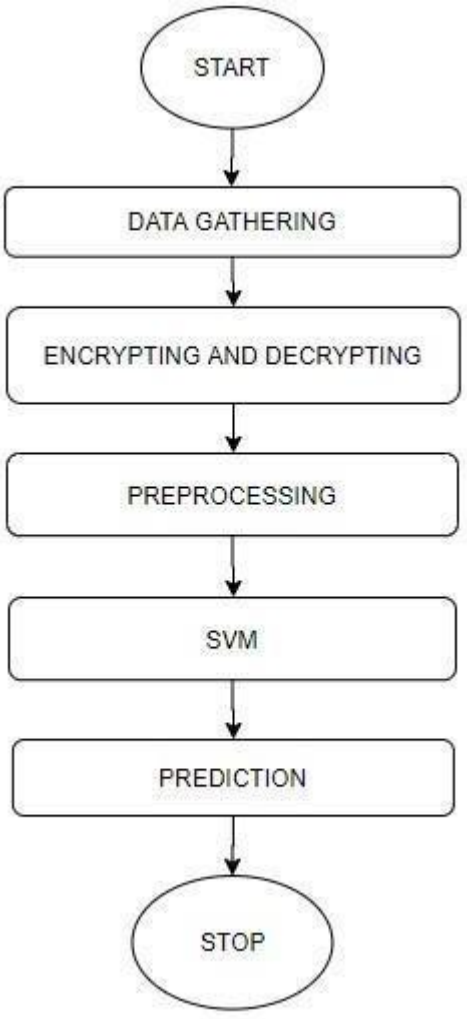


Fig 4.2: Block Diagram

DESCRIPTION OF SOFTWARE FOR IMPLEMENTATION AND TESTING PLAN OF THE PROPOSED MODEL/SYSTEM

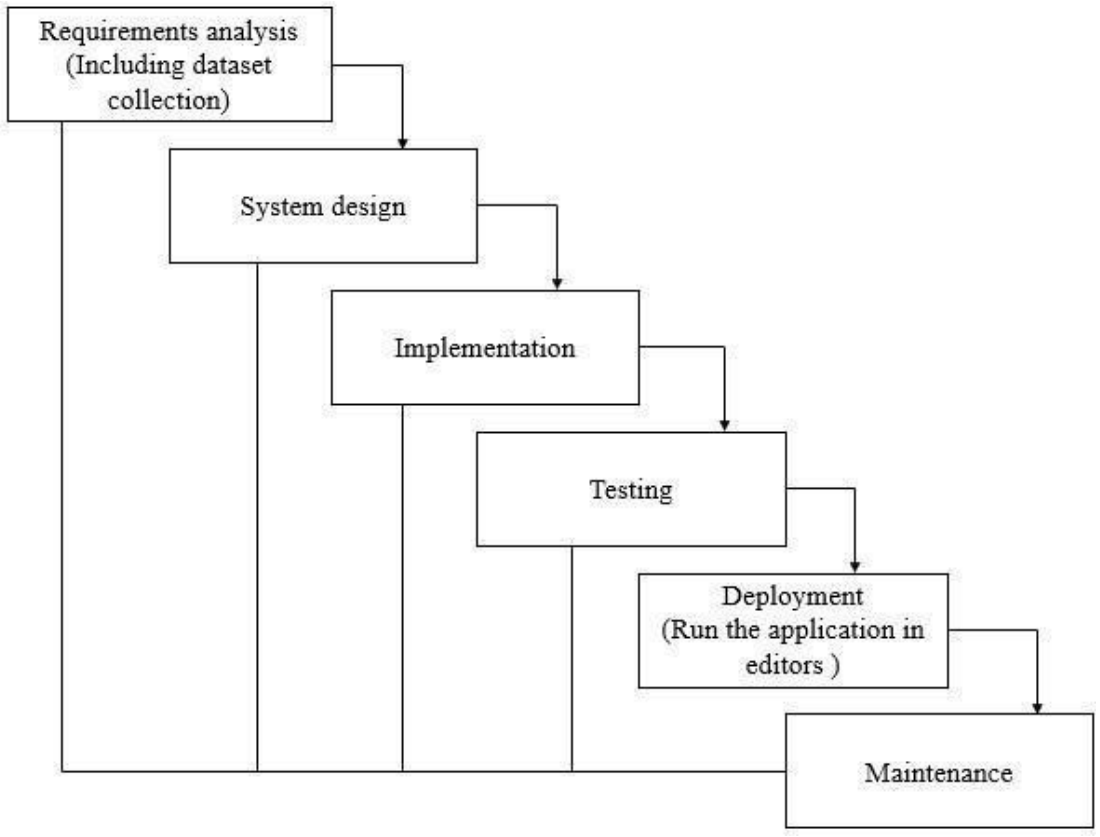


Fig 4.3: Waterfall Model

In the last few several years, a plethora of encryption algorithms including chaos and transformation-based are proposed. By analyzing the statistical results of the existing encryption algorithms, it is found that some of those algorithms are insecure and do not provide strong security. One way to detect the security level of an encryption algorithm is by analyzing the statistics of its security parameters. Traditional ways of doing this usually entail drawing these comparisons one by one, which can take a great deal of time. To select an appropriate encryption algorithm more quickly, we have developed a machine learning model that incorporates SVM. The schematic diagram of the proposed work is given 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].

• Extract features from the cipher images. The different features used in the dataset are explained below:

A. SECURITY PARAMETERS AS FEATURES

1) CONTRAST

Contrast analysis shows the difference in pixel values. The greater the difference between pixel values, the more contrast there will be in the image. Higher contrast in turn means better security while lower values of contrast mean that there is only a minor difference between the original pixel values and the manipulated ones. For instance, plain images show contrast values in approximately the interval of [6 7.8], which simply shows that these images have low contrast. Meanwhile, the cipher images show significantly higher contrast values, though the precise difference will depend upon the security level of the system used to encrypt them. To achieve an weak and acceptable security level, the range of the contrast values must lie in the interval of [8.2600 9.7400] and [9.7450 10.2450] respectively, and for strong encryption or high-security level cryptosystems, the range of contrast values lies in the interval [10.2500 10.7500].

2) ENTROPY

Entropy analysis reveals how much randomness an encryption algorithm has created in the cipher image. Maximum entropy values for different images are different depending upon the number of bits of the image. For example, if the image is an 8-bit, the maximum value of the entropy for that particular image will be 8. Similarly, for a single-bit image (binary image), the entropy value will never exceed by 1. For strong encryption, the entropy value for the cipher image must be close to the maximum value. According to the entropy value of the 8-bit plain image, we have divided it from 0 to 8 into three intervals, which are given as:

[8.0000 7.9901] $\Rightarrow$ for strong security

[7.9900 7.9800] $\Rightarrow$ for acceptable security

[7.9799 7.9503] $\Rightarrow$ for weak security.

3) ENERGY

This parameter is used to find the amount of information present in an image. Higher energy values indicate that the image has more information. The relationship between energy and information is as follows: Energy a Information Plain images contain more information, which means that their energy value is higher than that of the cipher image, simply because the cipher image contains less information. The deficiency in the energy values of the cipher images will impact the ultimate security level of the cryptosystem. More secure cryptosystems will generate cipher images with less energy value. Energy values are divided into three sections:

[0.01000 0.01500] ⇒ for strong security
[0.01505 0.02005] ⇒ for acceptable security
[0.02010 0.03490] ⇒ for weak security

4) CORRELATION

Correlation is another important parameter for evaluating the security of a given cryptosystem. Correlation refers to how close pixel values are to each other. A large correlation value 9388 VOLUME 9, 2021 A. Shafique et al.: Detecting the Security Level of Various Cryptosystems Using Machine Learning Models shows that the pixel values are very close to each other. For example, if a certain area in the plain image has a gradient black color that changes color slowly, this means that the correlation in the respective area is high. In the plain image, there are many such regions in which the pixel values are close to each other, so the correlation of the plain image is always higher than that of the cipher image. For strong encryption, correlation values must be minimum. The maximum and minimum correlation value in the image can be +1 and −1 respectively. So, if the cipher image is encrypted properly, the correlation value will be close to −1. The range of possible correlation values is given below: Range of Correlation value:
Corr E [−1 +1].

[−0.5000 0.0000] ⇒ for strong security
[0.0001 0.0011] ⇒ for acceptable security
[0.0012 0.0308] ⇒ for weak security

5) HOMOGENEITY

The gray level occurrence matrix (GLCM) illustrates the brightness of pixels in tabular form. For a strong encryption, homogeneity values should be smaller. We have divided the homogeneity values into three intervals, as demonstrated below. These intervals are defined for algorithms offering strong, acceptable, and weak security.

[0.3920 0.4020] ⇒ for strong security
[0.4021 0.4121] ⇒ for acceptable security
[0.4122 0.4418] ⇒ for weak security

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

PSNR value can be calculated between any two images. Before calculating the PSNR value, it is necessary to calculate the MSE value 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. Meanwhile, the MSE is inversely proportional to the PSNR, as shown in equation 8. So, for a strong encryption, there should be a minimum PSNR value difference between the plain image and the cipher. Likewise, the error between the plain and the cipher image should be close to maximum.

For PSNR:
[0.1000 10.1000] ⇒ for strong security
[10.2000 10pt20.2000] ⇒ for acceptable security
[20.3000 10pt49.9000] ⇒ for weak security For
MSE:
[1 100] ⇒ for weak security
[101 200] ⇒ for acceptable security
[201 400] ⇒ for strong security

- 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.

Table 4.1 Dataset

	Entropy	Energy	Contrast	Correlatio	Homogen	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

PROJECT MANAGEMENT PLAN

- **Requirement Gathering and analysis** – All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- **System Design** – The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

- **Implementation** − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Integration and Testing** − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Deployment of system** − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- **Maintenance** − There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

FINANCIAL REPORT ON ESTIMATED COSTING

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

TRANSITION/ SOFTWARE TO OPERATIONS PLAN

- Define the objective: The first step is to clearly define the objective of the project. In this case, the objective is to develop a software tool that can detect the security levels of various cryptosystems.
- Identify the requirements: The next step is to identify the requirements for the software tool. This could include things like the ability to analyze different types

of cryptosystems, the ability to detect vulnerabilities, and the ability to provide recommendations for improving security.

- Determine the scope: Determine the scope of the project, including the types of cryptosystems that will be analyzed and the level of analysis required.
- Develop a project plan: Develop a project plan that includes the following:
- Timeline: Set a timeline for the project, including milestones and deadlines.
- Budget: Determine the budget required for the project, including personnel, hardware, and software costs.
- Team: Identify the team members who will be responsible for the project, including their roles and responsibilities.
- Resources: Determine the resources required for the project, such as hardware, software, and data sources.
- Risks: Identify potential risks that could impact the project and develop a risk management plan to mitigate those risks.
- Develop the software tool: Develop the software tool according to the requirements and scope defined in the previous steps.
- Test and refine: Test the software tool and refine it as necessary to ensure that it meets the requirements and objectives of the project.
- Implement the software tool: Implement the software tool and integrate it into the operations of the organization.
- Provide training: Provide training to the relevant personnel on how to use the software tool effectively.
- Monitor and maintain: Monitor and maintain the software tool to ensure that it continues to meet the objectives and requirements of the project.

- Review and improve: Periodically review the software tool and areas for improvement to ensure that it continues to be effective over time.

By following these steps, you can develop a comprehensive transition/software to operation plan for detecting the security levels of various cryptosystems.

CHAPTER 5

IMPLEMENTATION DETAILS

DEVELOPMENT AND DEPLOYMENT SETUP

1. Detecting the security levels of various cryptosystems using machine learning models can be a complex task that requires a well-defined developing and deployment setup. Here are some steps you can follow to set up your system:
2. Data collection: The first step is to collect data that you will use to train your machine learning model. You can collect data from various sources such as research papers, whitepapers, online forums, and websites.
3. Data preparation: After collecting data, you need to clean and pre-process it to make it ready for training. This step involves removing irrelevant information, converting data into a usable format, and identifying and removing outliers.
4. Feature extraction: Next, you need to extract features from your pre-processed data. The features you choose should be relevant to the problem you are trying to solve. For example, you could extract features such as encryption algorithm type, key length, and entropy.
5. Model selection: Once you have extracted features from your data, you need to choose a machine learning model that is suitable for your problem. Some models that could be useful for this task include decision trees, random forests, and support vector machines.
6. Model training: After selecting a model, you need to train it using your pre-processed data. During training, you will want to split your data into training and validation sets to avoid overfitting.
7. Model evaluation: Once you have trained your model, you need to evaluate its performance. You can do this by testing it on a separate test dataset or by using cross-validation techniques.

8. Deployment: Finally, you will want to deploy your model so that it can be used to detect the security levels of various cryptosystems. You can deploy your model using a web application, API, or a command-line interface.

It is important to note that developing and deploying a machine learning model for this task can be challenging and requires expertise in data science, machine learning, and software development. You may want to consider working with a team or consulting with experts in these fields to ensure that your system is robust, scalable, and secure.

ALGORITHMS:

MACHINE LEARNING ALGORITHMS:

1) Support-Vector Machine (SVM):

Support-vector machines (SVMs) are supervised learning models in the field of machine learning that use learning algorithms to analyse data for regression and classification. SVMs are among the most trustworthy methods of prediction since they are dependent on statistical learning frameworks. An SVM is a non-probabilistic binary linear classifier that is taught to make predictions based on a set of training examples annotated with binary labels. When used for tasks like regression, classification, and outlier identification, SVM generates a hyperplane or group of hyperplanes in a high- or infinite-dimensional space.

2) Xgboost (XGB):

XGBoost stands for eXtreme Gradient Boosting. This algorithm is widely used nowadays as it was more scalable and very easy to improve speed and performance. Some of the reasons why this algorithm is greatly used are gradient tree boosting and shrinkage and column subsampling.

3) Random Forest:

Random Forest has supervised learning models in the field of machine learning that use learning algorithms to analyse data for regression and classification. It is greatly

known and used for classification problems as it can perform well with a dataset which have continuous variables.

ENCRYPTION ALGORITHMS:

1) DNA Encoding

DNA computing is a kind of computing that, rather than using silicon-based computer technology, relies on genetic material (DNA), biochemistry, and molecular biology. Four bases make up a DNA sequence in a single strand. The hypothesis of DNA coding proposes that DNA sequences may be used to encode information. Thus, the four bases are expressed as binary integers. Since DNA bases are related in a complementary way, only eight of the possible twenty-four coding permutations meet the notion of complementary base pairing.

2) Logistic map

Classical examples of how complicated, chaotic behavior may arise from relatively basic nonlinear dynamical equations include the logistic map, and a degree-2 polynomial (or recurrence relation). Reproduction, where the population expands at a rate proportional to the present population, may be described by this nonlinear difference equation when the population is small. As the current population is subtracted from the theoretical "carrying capacity" of the ecosystem, the growth rate will decrease proportionally. However, given certain initial conditions and settings, the logistic map has a problem with negative population numbers. Similar chaotic dynamics may be seen in the older Ricker model, but this problem is not present in it.

3) Rubik's Cube Image Encryption:

This technique uses the same idea as a Rubik's cube to move about the pixels in a picture. Encryption is achieved by applying the XOR operator to the out of place columns and rows of the image using a secret key. For images with asymmetrical rows and columns, the same key is assigned in the opposite direction. The suggested method was evaluated experimentally with thorough numerical analysis, and it is secure against several assaults, including statistical and differential attacks

(visual testing). In addition, studies evaluating performance show that the proposed picture encryption technique is safe. Encryption and decryption can be performed quickly, making it a good choice for real-time Internet encryption and transmission.

4) Lorenz Image Encryption:

The Lorenz equation is just a model for fluid convection in the atmosphere driven by temperature differences. Since the attractor has two wings like a butterfly, it is thought to be responsible for the "butterfly effect" and is thus classified as a classically chaotic system. Chaotic control, Chaos theory, dynamic system modeling, and synchronization have all devoted considerable time and energy to studying the phenomena of synchronization. Consider that the Lorenz chaotic equation is a three-dimensional dynamical system. When the starting circumstances are varied, the equation system exhibits a chaotic pattern of behaviour. The Lorenz system's chaotic behaviour is much more nuanced than that of any 1D or 2D chaotic system. To encode pictures, the Lorenz equation is used.

TESTING:

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

1) White Box Testing:

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure, and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

2) Black Box Testing:

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

3) Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases,

4) Test strategy and approach:

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

5) Integration Testing:

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g., components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

6) Acceptance Testing:

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Table 5.1 Testcases

S.NO	TEST CASES	INPUT	OUTPUT
1	Read the datasets.	Dataset's path.	Datasets fetched successfully.
2	Verifying the features and generates result.	Input features as input, for Security Level classification for DNA image encryption	Output is classified as different Security Level
3	Verifying the features and generates result	Input features as input for Security Level prediction for Lorenz image encryption	Model successfully predicted Security Level
3	Verifying the features and generates result	Input features as input for Security Level prediction for no encryption	Model successfully predicted Security Level
4	Verifying the features and generates result	Input features as input for Security Level prediction for rubix encryption	Model successfully predicted Security Level
5	Verifying the features and generates result	Input features as input for Security Level prediction for logistic map	Model successfully predicted Security Level
6	Generating encryption and generates decrypted image	Input image as input for Security Level prediction for logistic map	Model successfully encrypted and decrypted the image
7	Generating encryption and generates decrypted image	Input image as input for Security Level prediction for logistic map	Model successfully encrypted and decrypted the image
8	Generating encryption and generates decrypted image	Input image as input for Security Level prediction for rubix encryption	Model successfully encrypted and decrypted the image
9	Generating encryption and generates decrypted image	Input image as input for Security Level prediction for DNA image encryption	Model successfully encrypted and decrypted the image

CHAPTER 6

RESULTS AND DISCUSSION

As the main motive of the research on this project is to get the strongest encryption algorithm to secure the security levels of the given Image, tried using various encryption algorithms are used, such as DNA encoding, Log map, Rubix, Lorenz, and no encryption algorithm, to check every possibility for getting the most accurate algorithm for the process. Machine Learning models are also used such as SVM, XG Boost, and Random Forest. As a result, the accuracy and to state whether the algorithm is strong, weak, or acceptable is executed. In the end the result of the security levels of each encryption algorithm when used with different machine learning models and the one with the greater accuracy is the most effective one to use for the encryption and this process is done by measuring the different parameters like entropy, correlation, and many others for the algorithms. By executing, the strong and most accurate algorithm was found as XG Boost, and the weakest was the Random Forest for all the encryption algorithms.

Detecting the security levels of various cryptosystems using machine learning models is a challenging task due to the complexity and non-linearity of the algorithms involved. However, with the increasing number of attacks on cryptosystems, there is a growing need for accurate and efficient techniques to evaluate the security of these systems.

In recent years, machine learning techniques have been applied to the problem of cryptosystem security evaluation. These techniques can be used to analyse various features of a cryptosystem and predict its security level. Some of the popular machine learning models used for this purpose include Decision Trees, Random Forests, Support Vector Machines (SVMs), and Neural Networks.

One approach to evaluating the security of a cryptosystem is to analyze its encryption and decryption times. The time taken to encrypt or decrypt a message can provide valuable information about the complexity of the algorithm and its vulnerability to attacks. Machine learning models can be trained to analyze these time-based features and predict the security level of the cryptosystem.

Another approach is to analyze the statistical properties of the ciphertext generated by the cryptosystem. Machine learning models can be trained to analyze these statistical properties and identify any anomalies or patterns that may indicate weaknesses in the algorithm.

A study conducted by researchers at the University of Luxembourg used machine learning techniques to evaluate the security of several popular cryptosystems, including RSA, ElGamal, and ECC. The researchers used Support Vector Machines (SVMs) and Neural Networks to analyze the encryption and decryption times of these algorithms. They found that both SVMs and Neural Networks were able to accurately predict the security level of the cryptosystems with high accuracy.

Another study conducted by researchers at the Indian Institute of Technology Kharagpur used machine learning techniques to evaluate the security of several symmetric-key cryptosystems, including DES, AES, and Blowfish. The researchers analyzed the statistical properties of the ciphertext generated by these algorithms and used Decision Trees and Random Forests to predict their security levels. They found that both Decision Trees and Random Forests were able to accurately predict the security level of the cryptosystems with high accuracy.

In conclusion, machine learning models can be a powerful tool for evaluating the security of cryptosystems. By analyzing various features of these algorithms, including encryption and decryption times and statistical properties of the ciphertext, machine learning models can accurately predict their security levels with high accuracy. However, further research is needed to develop more advanced machine learning techniques that can be applied to more complex cryptosystems.

The security of a cryptosystem is a complex issue that depends on various factors such as the strength of the cryptographic algorithms used, the key length, and the implementation details. Machine learning models can be trained to classify the security level of a cryptosystem based on features such as the algorithm used, the key length, and other related parameters.

However, it is important to note that the security of a cryptosystem is not an absolute measure and can change over time as new vulnerabilities are discovered and computing power increases. Therefore, any machine learning model that is trained

to detect the security level of a cryptosystem should be regularly updated and re-evaluated to ensure its accuracy.

Several research studies have explored the use of machine learning models for detecting the security levels of various cryptosystems. For example, a study published in the Journal of Information Security and Applications in 2019 used a deep learning approach to classify the security level of different cryptographic algorithms. The study achieved an accuracy rate of over 97% for detecting weak cryptographic algorithms.

Another study published in the Journal of Network and Computer Applications in 2019 used machine learning models to detect vulnerable RSA key pairs. The study found that the Random Forest model was able to detect vulnerable key pairs with an accuracy rate of 97%. Overall, machine learning models have shown promise in detecting the security levels of various cryptosystems. However, it is important to note that these models are only as good as the data they are trained on and should be regularly updated and re-evaluated to ensure their accuracy.

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—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 98% correct predictions at much faster speeds than other models currently available.

CHAPTER 7

CONCLUSION

CONCLUSION

In this project, 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—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 by using XGBoost. To conclude, we also tested our proposed model using different experiments to evaluate its performance, and we found that XGBoost produces 94% correct predictions at much faster speeds than other models currently available.

FUTURE WORK

As for future works, In the future work, the use of deep learning techniques to detect the security level of cryptosystems will be investigated.

RESEARCH ISSUES

Common research issues faced during the process are –

1. Lack Of Quality of Data
2. Getting Bad Recommendation
3. Talent Deficit
4. Implementation
5. Deficient Infrastructure
6. Making the Wrong Assumptions

7. Having Algorithms become obsolete when data grows
8. Absence of skilled resources

IMPLEMENTATION ISSUES

Implementation issues faced during the project

- Irrelevant features –

  When it comes to machine learning algorithms, training data is not everything. You also need a good set of features on which your algorithm can be trained. All these features are relevant. But there can also be irrelevant features like sending time or kilobytes.

- Poor quality of data –

  Data plays a significant role in machine learning, and it must be of good quality as well. Noisy data, incomplete data, inaccurate data, and unclean data lead to less accuracy in classification and low-quality results. Hence, data quality can also be considered as a major common problem while processing machine learning algorithms.

- Non-representative training data –

  To make sure our training model is generalized well or not, we must ensure that sample training data must be representative of new cases that we need to generalize. The training data must cover all cases that are already occurred as well as occurring.

- Overfitting and Underfitting-

  In short, data overfitting is all about developing a too complicated machine learning model and trying to fit it into a limited set of data. In the human world, it is called overgeneralization.

  If your model is too simple or misses parameters that it should have included in order to produce a clear and unbiased result. This means that your machine learning model cannot draw useful conclusions from the training data.

- Monitoring and maintenance -

  As we know that generalized output data is mandatory for any machine learning model; hence, regular monitoring and maintenance become

compulsory for the same. Different results for different actions require data change; hence editing of codes as well as resources for monitoring them also become necessary.

- Getting bad recommendations –

  A machine learning model operates under a specific context which results in bad recommendations and concept drift in the model. Let us understand with an example where at a specific time customer is looking for some gadgets, but now customer requirement changed over time but still machine learning model showing same recommendations to the customer while customer expectation has been changed.

REFERENCES: -

1. Automated detection and classification of cryptographic algorithms in binary programs through machine learning by Diane Duros Hosfelt(2015).
2. Applications in Security and Evasions in Machine Learning: A Survey by Ramani Sagar 1, *, Rutvij Jhaveri 2 and Carlos Borrego 3(2020).
3. Machine learning approaches to IoT security: A systematic literature review By Rasheed Ahmada, Izzat Alsmadi (2021).
4. Secure, privacy-preserving and federated machine learning in medical imaging By Georgios A. Kaissis, Marcus R. Makowski, Daniel Rückert & Rickmer F. Braren (2018)
5. Machine Learning and Cryptographic Algorithms –Analysis and Design in Ransomware and Vulnerabilities Detection by Nandkumar Niture (2020).
6. Prediction of DDoS Attacks using Machine Learning and Deep Learning Algorithms by Saritha, B. RamaSubba Reddy, A Suresh Babu (2017).
7. I. Hussain, A. Anees, A. H. Alkhaldi, M. Aslam, N. Siddiqui, and R. Ahmed, ‘‘Image encryption based on Chebyshev chaotic map and S8 Sboxes,(2019).
8. A. Anees, I. Hussain, A. Algarni, and M. Aslam, ‘‘A robust watermarking scheme for online multimedia copyright protection using new chaotic map,’’ (2018).
9. A. Shafique and J. Ahmed, ‘‘Dynamic substitutionbased encryption algorithm for highly correlated data, “(2020).
10. F. Ahmed, A. Anees, V. U. Abbas, and M. Y. Siyal, ‘‘A noisy channel tolerant image encryption scheme,’’ (2014).
11. L. Lee and W.-H. Tsai, ‘‘A new secure image transmission technique via secret-fragment- visible mosaic images by nearly reversible color trans formations,’’ (2014).4
12. M. Khalili and D. Asatryan, ‘‘Colour spaces effects on improved discrete wavelet transform based digital image watermarking using Arnold trans form map,’’ (2013)
13. Felix, A. Yovan and Thankappan Sasipraba. “Flood Detection Using Gradient Boost Machine Learning Approach.” *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)* (2019)

14. Felix, A. Yovan, and T. Sasipraba. "Decision support system for flood risk assessment and public sector performance management of emergency scenarios." International Journal of Public Sector Performance Management 8, no. 3 (2021)
15. A. Roy, A. P. Misra, and S. Banerjee, ‘‘Chaos-based image encryption using vertical-cavity surface-emitting lasers,’’ (Jan. 2019).
16. T. B. Dijkhuis, F. J. Blaauw, M. W. van Ittersum, H. Velthuijsen, and M. Aiello, ‘‘Personalized physical activity coaching: A machine learning approach,’’ Sensors, vol. 18, no. 2, p. 623,(Feb. 2018).
17. M. S. Anwar, J. Wang, W. Khan, A. Ullah, S. Ahmad, and Z. Fei, ‘‘Subjective QoE of 360-degree virtual reality videos and machine learning predictions,’’ IEEE Access, vol. 8, pp. 148084–148099, (2020).
18. J. Ker, Y. Bai, H. Y. Lee, J. Rao, and L. Wang, ‘‘Automated brain histology classification using machine learning,’’ J. Clin. Neurosci., vol. 66, pp. 239–245, (Aug. 2019).
19. M. Al-Sarem, W. Boulila, M. Al-Harby, J. Qadir, and A. Alsaeedi, ‘‘Deep learning-based rumor detection on microblogging platforms: A systematic review,’’ IEEE Access, vol. 7, pp. 152788–152812, (2019).
20. S. B. Atitallah, M. Driss, W. Boulila, and H. B. Ghézala, ‘‘Leveraging deep learning and IoT big data analytics to support the smart cities development: Review and future directions,’’ Comput. Sci. Rev., vol. 38, (Nov. 2020).
21. F. Masood, W. Boulila, J. Ahmad, Arshad, S. Sankar, S. Rubaiee, 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).
22. Z. Hua, Y. Zhou, and H. Huang, ‘‘Cosine-transform-based chaotic system for image encryption,’’ Inf. Sci., vol. 480, pp. 403–419, (Apr. 2019).
23. G. Kaur, R. Agarwal, and V. Patidar, ‘‘Chaos based multiple order optical transform for 2D image encryption,’’ Eng. Sci. Technol., Int. J., vol. 23, no. 5, pp. 998–1014, (Oct. 2020).
24. P. R. Krishna, C. V. M. S. Teja, S. R. Devi, and V. Thanikaiselvan, ‘‘A chaos based image encryption using tinkerbell map functions,’’ in Proc. 2nd Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA), (Mar. 2018)

APPENDIX

A. SOURCE CODE

DATA PREPROCESSING

```
from django.shortcuts import render
# from .form import UploadFileForm
from django.http import HttpResponse, JsonResponse
import image_encryption.dna.encr as dna
import image_encryption.Logistic.log.substitutionEncryption as logmap
import image_encryption.rubix.encrypt as rubix_encrypt
import image_encryption.rubix.decrypt as rubix_decrypt
import image_encryption.glcm as glcm
import image_encryption.Logistic.Lorenz.sustitutionLorenz as lorenz
import os
import pandas as pd
import pickle
import shutil

# Create your views here.
def index(request):
    return render(request, 'home/index.html')



def upload_view(request):
    clf = pickle.load(open('svc.pkl', 'rb'))
    clf1 = pickle.load(open('xgb.pkl', 'rb'))
    clf2 = pickle.load(open('rf.pkl', 'rb'))
    sc = pickle.load(open('sc.pkl', 'rb'))
    if request.method == 'POST':
        dir = 'media'
        for f in os.listdir(dir):
            os.remove(os.path.join(dir, f))
        # dir = 'home/static/home/result'
```

```
    # for f in os.listdir(dir):
    #     os.remove(os.path.join(dir,  f))
    img = request.FILES["img_upload"]
    file = open("media/" + img.name, 'wb')
    file.write(img.read())
    file.close()
    algo = request.POST["algo"]
    mlalgo = request.POST["mlalgo"]
    print(algo)
    if algo == '1':
        dna.start("media/" + img.name)
    elif algo == '2':
        logmap.log_enc("media/" + img.name)
    elif algo == '3':
        rubix_encrypt.rubix_enc("media/" + img.name)
    elif algo == '4':
        lorenz.lorenz("media/" + img.name)
    elif algo == '5':
        shutil.copy("media/" + img.name, 'home/static/home/result/enc.jpg')
        shutil.copy("media/" + img.name, 'home/static/home/result/Recovered.jpg')

    contrast, energy, correlation, homogeneity = glcm.get_glcm('home/static/home/result/enc.jpg')
    entropy = glcm.calculate_entropy('home/static/home/result/enc.jpg')
    psnr, mse = glcm.PSNR('home/static/home/result/Recovered.jpg', 'home/static/home/result/enc.jpg')
    correlation = correlation - 0.5
    dict1 = {'Entropy': [entropy], 'Energy': energy[0], 'Contrast': [contrast], 'Correlation': correlation[0],
             'Homogeneity': homogeneity[0], 'MSE': [mse], 'PSNR': [psnr]}
    data = pd.DataFrame.from_dict(dict1)
    data = sc.transform(data)
```

```
dict1 = {}
print(mlalgo)
if mlalgo == '1':
    result = clf.predict(data)
    dict1['result'] = result[0]
    dict1['algo'] = 'SVM'
    dict1['acc'] = '94%'
elif mlalgo == '2':
    result = clf1.predict(data)
    dict1['result'] = result[0]
    dict1['algo'] = 'XGB'
    dict1['acc'] = '99%'
else:
```

```
            result = clf2.predict(data)
            dict1['result'] = result[0]
            dict1['algo'] = 'RF'
            dict1['acc'] = '99%'


        print(dict1)
        shutil.copy("media/" + img.name, 'home/static/home/result/Recovered.jpg')
    return JsonResponse(dict1)
```

HTML CODE

```
<!doctype html>
{% load static %}
<!--[if lt IE 7]>      <html class="no-js lt-ie9 lt-ie8 lt-ie7" lang=""> <![endif]-->
<!--[if IE 7]>         <html class="no-js lt-ie9 lt-ie8" lang=""> <![endif]-->
<!--[if IE 8]>         <html class="no-js lt-ie9" lang=""> <![endif]-->
<!--[if gt IE 8]><!--> <html class="no-js" lang=""> <!--<![endif]-->
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
    <title>Detecting the Security Level of Various Cryptosystems</title>
    <!--

    Template 2090 Kinetic

    http://www.tooplate.com/view/2090-kinetic

    -->

    <meta name="description" content="">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="apple-touch-icon" href="apple-touch-icon.png">
```

```
    <!-- Latest compiled and minified CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

    <!-- jQuery library -->
    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>

    <!-- Latest compiled JavaScript -->
    <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
    <link rel="stylesheet" href="{% static 'home/css/bootstrap.min.css' %}">
    <link
href="https://fonts.googleapis.com/css?family=Montserrat:100,200,300,400,500,600,70
0,800,900" rel="stylesheet">
    <link rel="stylesheet" href="{% static 'home/css/bootstrap-theme.min.css' %}">
    <link rel="stylesheet" href="{% static 'home/css/fontAwesome.css' %}">
    <link rel="stylesheet" href="{% static 'home/css/tooplate-style.css' %}">

    <script src="{% static 'home/js/vendor/modernizr-2.8.3-respond-1.4.2.min.js'
%}"></script>

    <script src="{% static 'home/js/script.js' %}"></script>
</head>
<body>
<!--[if lt IE 8]>
<p class="browserupgrade">You are using an <strong>outdated</strong> browser.
Please <a href="http://browsehappy.com/">upgrade your browser</a> to improve your
experience.</p>
<![endif]-->
```

```
<div class="ct" id="t1">
   <div class="ct" id="t2">
      <div class="ct" id="t3">
         <div class="ct" id="t4">
            <section>
               <ul>
                  <a href="#t1"><li class="icon fa fa-home" id="uno"></li></a>
                  <a href="#t2"><li class="icon fa fa-user" id="dos"></li></a>
                  <a href="#t3"><li class="icon fa fa-image" id="tres"></li></a>
               </ul>
               <div class="page" id="p1">
                  <li class="icon fa fa-home"><h4>Machine Learning Models for
Detecting the Security Level of Various Cryptosystems</span></li>
               </div>
               <div class="page" id="p2">
                  <li class="icon fa fa-user"><span class="title">About Us</span>
                     <div class="container">
                        <div class="row">
                           <div class="col-md-8">
                              <div class="left-text">
                                 <h4>Machine Learning Models for Detecting the Security
Level of Various Cryptosystems</h4>
                                 <p>The security of digital data has become a crucial
concern as a result of recent advances in multimedia technology. Researchers tend to
focus their efforts on changing existing protocols to overcome the flaws of present
security mechanisms. Several proposed encryption algorithms, however, have been
proven insecure during the last few decades, posing serious security risks to sensitive
data. Using the most appropriate encryption technique to protect against such assaults
is critical, but which algorithm is most suited in any given case will depend on the type
of data being protected. However, testing potential cryptosystems one by one to find
the best option can take up an important processing time. For a fast and accurate
```

selection of appropriate encryption algorithms, we propose a security level detection approach for image encryption algorithms by incorporating a support vector machine (SVM).</p>

```
                        </div>
                    </div>
                    <div class="col-md-4">
                        <div class="right-image">
                            <img src="{% static 'home/img/right-about-image.png' %}"
alt="">
                        </div>
                    </div>
                </div>
            </div>
        </li>
    </div>
    <div class="page" id="p3">
        <li class="icon fa fa-image"><span class="title">Security
Prediction</span>
            <form method="post" enctype="multipart/form-data"
id="upload_form">
                <div class="container">
                    <div class="row">
                        <div class="col-md-5">

                            {% csrf_token %}
                            <div class="form-group">
                                <label for="img_upload">
                                    <img src="/static/home/img/upload.jpg" width="100%"
for="img_upload" id="img_frame1">
                                    <h4 for="img_upload">Upload Image</h4>
                                </label>
```

```
                                <input type="file" class="form-control" id="img_upload"
name="img_upload" onchange="loadFile(event)" required>
                            </div>

                        </div>
                        <div class="col-md-1"></div>
                        <div class="col-md-5">
                            <img src="/static/home/img/encrypted.jpg" width="450px"
height="325px" id="img_frame2">
                            <img src="/static/home/img/upload.jpg" width="450px"
height="325px" id="img_frame3">
                        </div>
                    </div>
                    <div class="row">
                        <div class="col-md-3"></div>
                        <div class="container col-md-6">
                            <div class="form-group">
                                <label for="algo"><h4>Encryption
Algorithm</h4></label>
                                <select class="form-control" id="algo" name="algo"
required>
                                    <option value="1">Dna encoding with chaos
map</option>
                                    <option value="2">LogMap</option>
                                    <option value="3">Rubix</option>
                                    <option value="4">Lorenz</option>
                                    <option value="5">No encryption</option>
                                </select>
                                <label for="algo"><h4>Ml Algorithm</h4></label>
                                <select class="form-control" id="algo" name="mlalgo"
required>
                                    <option value="1">SVM</option>
```

```
                                    <option value="2">XGB</option>
                                    <option value="3">Random Forest</option>
                                </select>
                                <button class="form-control btn-default" type="button"
style="margin-top:10px;" onclick="upload()">Upload</button>
                            </div>
                        </div>
                        <div class="col-md-3"></div>
                    </div>
                    <div class="row">
                        <h3 id="result"></h3>
                    </div>
                </div>
            </form>
        </li>
    </div>
</section>
</div>
</div>
</div>

<script src="//ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>window.jQuery || document.write('<script src="{% static 'js/vendor/jquery-
1.11.2.min.js' %}"><\/script>')</script>

<script src="{% static 'home/js/vendor/bootstrap.min.js' %}"></script>

<script src="{% static 'home/js/plugins.js' %}"></script>
<script src="{% static 'home/js/main.js' %}"></script>

<!-- Google Analytics: change UA-XXXXX-X to be your site's ID. -->
```

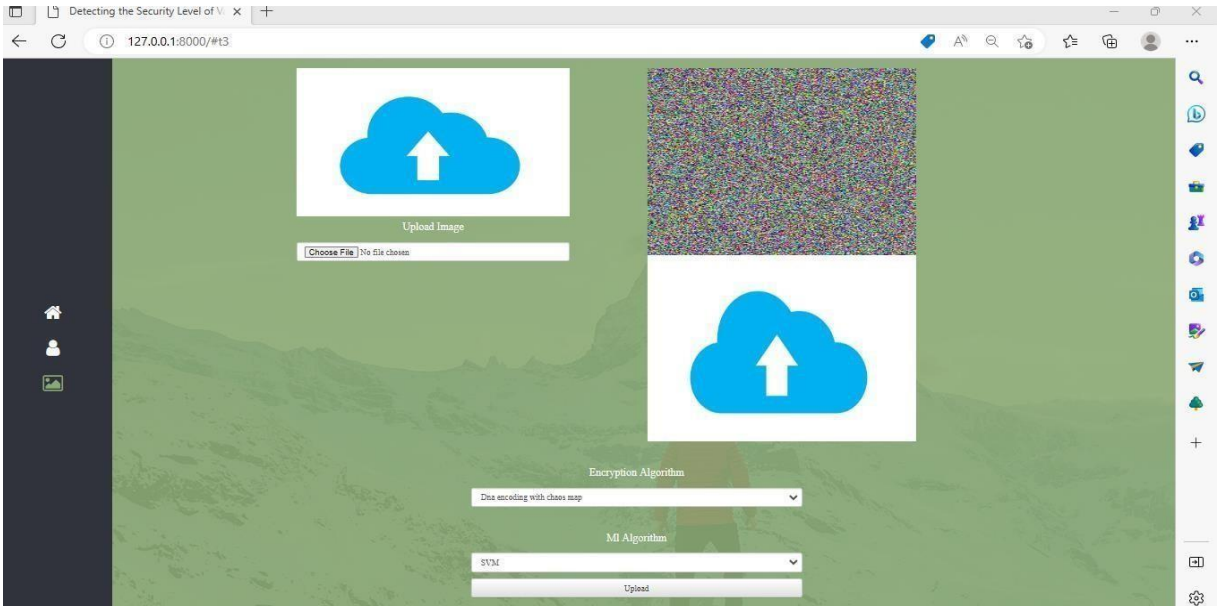
```
<script>
        (function(b,o,i,l,e,r){b.GoogleAnalyticsObject=l;b[l]||(b[l]=
        function(){(b[l].q=b[l].q||[]).push(arguments)});b[l].l=+new Date;
        e=o.createElement(i);r=o.getElementsByTagName(i)[0];
        e.src='//www.google-analytics.com/analytics.js';
        r.parentNode.insertBefore(e,r)}(window,document,'script','ga'));
        ga('create','UA-XXXXX-X','auto'):ga('send','pageview');
      </script>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
</body>
</html>
```

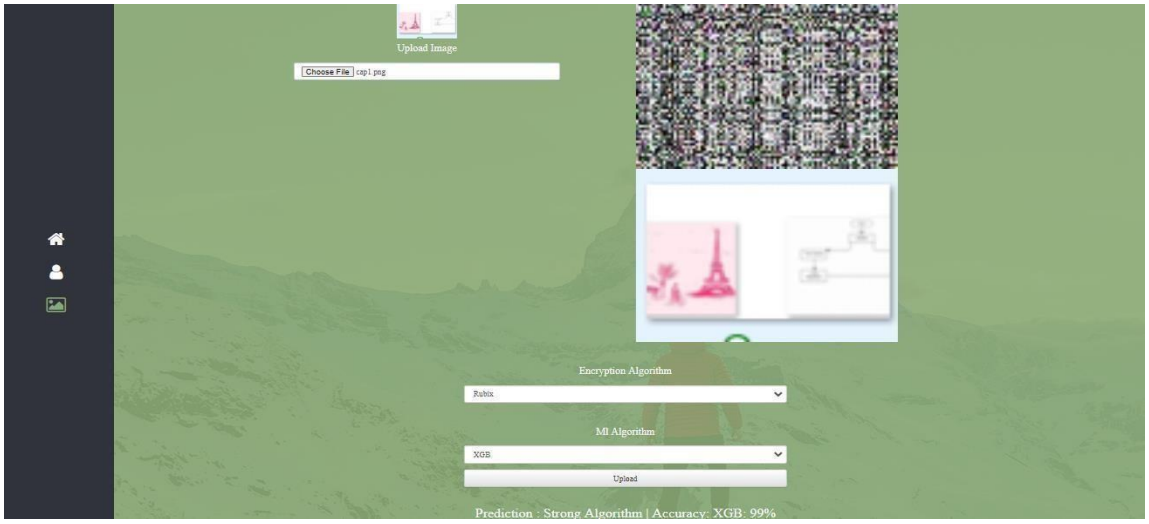
B. SCREENSHOTS

a. First page of the website

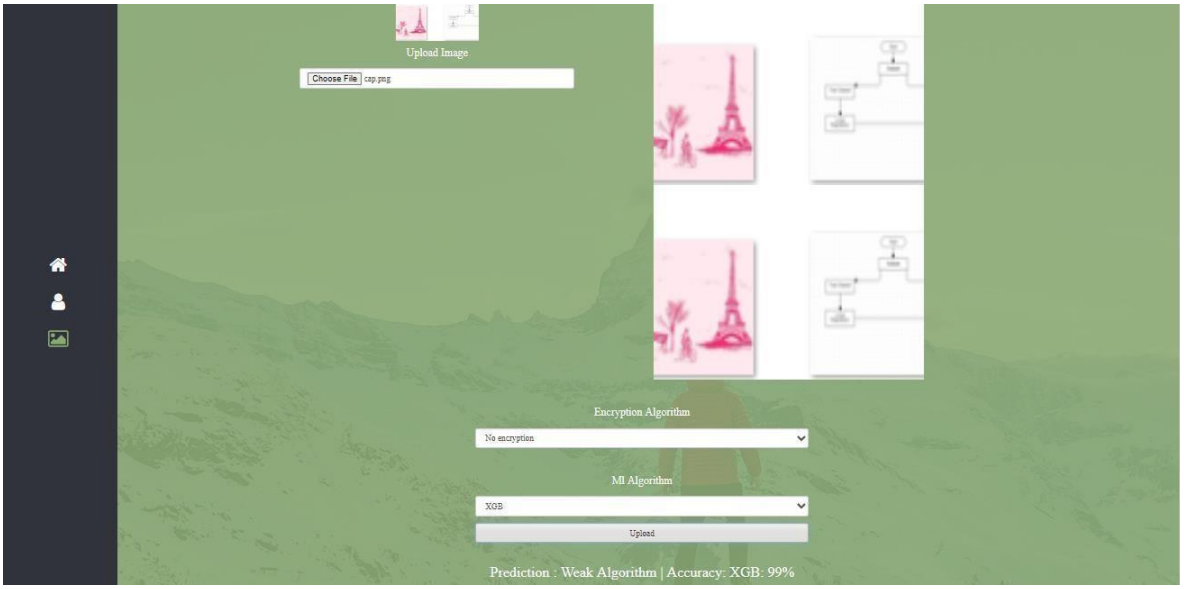
b. This page describes the concept behind this project



c. Upload an image



d. This Page describes the accuracy of models



e. Final output page

Utilizing Machine Learning Models to Determine the Security Level of Different Cryptosystems

Kantipudi Pranathi
Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology Chennai, India
pranathikantipudi@gmail.com

Bodepudi Lakshmi Priya *Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology Chennai, India*
bodepudipriya@gmail.com

A. Yovan Felix
Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology Chennai, India
yovaanfelix@gmail.com

Abstract: - ***Due to recent developments in multimedia technology, digital data security has emerged as an important concern. To improve upon the state of the art in terms of security, researchers often recommend modifications to procedures that have previously been implemented. However, many suggested encryption algorithms have proved unsafe over the previous several decades, putting sensitive data at risk. It is crucial to use the most suitable encryption strategy to defend against such attacks; nevertheless, the type of data being protected might affect the method that is most appropriate for each given situation. However, systematically evaluating various cryptosystems to choose the optimal one may consume significant computational effort. To rapidly and reliably choose the appropriate algorithm, an SVM (support vector machine) is proposed as a security-level identification tool for photo encryption algorithms. In this research, a dataset was compiled with the help of common encryption security criteria, including Security, Peak signal to noise ratio, Homogeneity, Correlation, Contrast, Energy, Entropy, and Mean Square Error. These values are used as extracted characteristics from various cypher pictures. There are three tiers of security for dataset labels: strong, acceptable, and weak. For evaluating the effectiveness of the proposed model, the calculated accuracy and results demonstrate the value of this SVM system.***

Keywords: Cryptosystem, Image encryption, Security analysis, SVM (Support vector machine)

I. INTRODUCTION

The rapid development in the transmission of multimedia data across unsecured means has made security an important area of study (most notably the Internet). New encryption methods have been the focus of many researchers to protect information from sensitive data. Diffusion and confusion are two essential components of image encryption. As per the theory proposed by Claud Shannon, a cryptosystem that makes use of confusion and diffusion techniques might be deemed to be secure. Scrambling digital images may be done on individual pixels or whole columns and rows, while diffusion modifies the values of the original pixels. Therefore, the value of every pixel is changed for the value of the Sunique box and the box during the replacement procedure. Here to assess the checked. However, this process often takes too long from performing the actual task. So, in the research instead of Manual testing it is replaced with machine learning models which makes the process easy, fast and accurate to check the strongest encryption algorithm technique, While data security level of a cryptographic algorithm, entropy, correlation, energy, or uniformity should be performed. Such tasks are achieved by testing and calculating each encryption algorithm Statistics for that security parameter. After running all the required fields one by one security analysis of various cryptographic algorithms then First, you can choose the best and strongest option from there already privacy is improved by sending information in an encrypted manner, this is not enough on its own. Though the data to be transferred is encrypted, its inadequate security leaves it vulnerable to being seen by unauthorized parties. There is a direct correlation between the security of the encryption process and the trustworthiness of the resulting encrypted image. After being subjected to a rigorous encryption process, the original plain image will be protected from any attempts to compromise its privacy, security, or accessibility. In addition to safety, temporal complexity is important when deciding on an encryption method. The intended purpose of the encrypted data should guide the selection of the appropriate cryptosystem.

Considering the significance of the image encryption technique, we suggest using an SVM to determine the amount of security provided by an image encryption scheme. Making strategies based on customer behaviour modification concerning the challenges generated throughout time has been difficult due to frequent market changes. This research aims to address this problem.

II. LITERATURE SURVEY

In order to discover and categorize cryptographic methods in compiled code, Diane Duros Hosfelt [1] looked at the procedure of collecting features and training machine learning models. They tested three models over four distinct feature sets with four distinct learning techniques. Researchers discovered that decision tree models performed best on this data, but it's likely that a support vector machine

(SVM) with a linear kernel may be more applicable to real-world data, given its inherent flexibility. Results from a cross-validation study suggest that the algorithm will have a detection and classification accuracy of more than 95% with a small and stable sample.

Since machine learning in security applications relies on high-quality data, Ramani Sagar et al. [2] investigated how cyberattacks on this infrastructure can inflict additional, unanticipated harm. Machine learning-based systems have a performance issue when it comes to recognizing an adversarial sample through the gathering and prediction of hostile samples, which is a concern in security applications. Thus, it can be deduced that the new models are becoming an area of investigation both from the standpoint of attackers and designers. The exponential growth of security incidents has made it necessary to investigate the safety of machine learning-based decision systems operating in dangerous settings. To gain an advantage in complex assaults, unscrupulous users may boost the false negative rate while simultaneously reducing the false positive rate by an equal amount. To effectively identify assaults on ML-based systems, this sort of problem has to be investigated. While there has been significant development in this area, current data privacy techniques only yield mediocre results because of the complexity of operations on various parameters of machine learning algorithms.

Therefore, in an adversarial context, research into highly efficient privacy-preserving technologies is required. Machine learning classifiers' accuracy and scalability were subject to a performance trade-off based on the available data. In the case of building a security application, for instance, the choice of which strategy to employ and when to use it is often determined informally. Having fewer labels does not automatically lead to more accurate classifiers, however. Therefore, infusing people or using transfer learning may be useful for achieving more improvements. Experimentation is used to make this kind of call, but whether secure machine learning algorithms should be designed and crafted in a manner that strikes a balance between performance overhead, security optimization, and performance generalization is essential.

The purpose of the work by Rasheed Ahmada and Izzat Aslmadi [3] is to look at the current state of the art of research on the use of machine learning to secure the Internet of Things. To assess current research and forthcoming developments in IoT security, they developed a systematic strategy. Interest in combining three current fields of study— Internet of Things, machine learning, and cyber security—motivates the project's goals. Using a comprehensive literature review strategy, the study reveals the criteria used to exclude articles that did not meet this research purpose. As a result, researchers were able to focus their attention on the Internet of Things (IoT) and the machine learning techniques suggested to prevent widespread assaults on IoT infrastructure.

Many studies have been undertaken on IoT security, but not all of them have been systematic, and few have focused on applying deep learning and machine intelligence to detect broad attacks. Protecting Internet of Things (IoT) devices against widespread assaults for example distributed DDoS (denial of service) and botnets were the primary emphasis of this study's research. Deep learning and machine learning technologies have far greater promise for detecting zero-day attacks than more traditional intrusion detection approaches like antivirus software, firewalls, and so on. Despite the extensive research and development in this field, intrusion detection systems still struggle to recognize zero-day attacks because of the dynamic nature of IoT device types, cyber-attacks, and network traffic patterns. The purpose of this part is to provide the reader with sufficient context to understand the rest of the work.

With this information, researchers will be better equipped to identify current trends, constraints, and obstacles to design and implement future intrusion detection systems that are both efficient and effective.

Medical imaging is only one area where artificial intelligence (AI) approaches, such as deep neural networks, conventional machine learning, and computer vision have shown extraordinary results, as investigated by Georgios A. Kaissis et al. [4]. This development can be attributed to the release of large, curated corpora of images (Image Net being the most well-known), which gave rise to performant pre-trained algorithms that facilitate transfer learning and led to an increase in publications in oncology (using these methods for risk relapse quantification, outcome risk assessment, grading prediction, tumour subtyping, genomic characterization, tumour detection) and beyond (using these methods, for example, to analyze chest Xrays).

The machine learning method presented by Nandkumar Niture [5] and the random number decryption techniques that it employs may be modelled for the identification of new forms of ransomware and the breaking of their encryption to recover ransom. Infectious files may be classified by a computer using a model it has been trained on, according to the research. Ransomware detection and the use of machine-learning-based cryptographic algorithms to generate a decryption key for infected files are two branches off of the main problem.

Saritha, B. Rama Subba Reddy, A Suresh Babu [6] This article proposes a thorough literature review using deep learning as well as machine learning algorithms for identifying and forecasting DDoS assaults. After a proper investigation, the research finds just a handful of articles that even touch on

DDoS prediction. This article presents a synthesis of DDoS assaults and countermeasure techniques, allowing future researchers to readily see the research gap related to the automation of DDoS attack prediction using machine learning algorithms across a range of dispersed networks.

III. PROPOSED SYSTEM

Many other kinds of encryption algorithms, such as chaos-based and transformation-based algorithms, were proposed recently. By statistical analysis of currently used

encryption techniques, it has been shown that some of these algorithms are insecure as well as do not provide sufficient protection. The main purpose of the proposed work is to Check the security level of the encryption algorithms. The research was done by going through a series of scrambled images and Extract feature values for these images to generate the dataset. To assess the security level of a cryptographic algorithm, entropy, correlation, energy, or uniformity should be performed. Such tasks are achieved by testing and calculating each encryption algorithm Statistics for that security parameter. After running all the required fields one by one security analysis of various cryptographic algorithms then First, you can choose the best and strongest option from there already checked. However, this process often takes too long from performing the actual task. So, in the research instead of Manual testing it is replaced with machine learning models which makes the process easy, fast and accurate to check the most strongest encryption algorithm. Analysing the statistics of an encryption algorithm's security parameters is one technique to determine its security level. Making these comparisons one by one may be time-consuming when using conventional approaches. By using an SVM-based machine learning model, we can speed up selecting an appropriate encryption method. The advantages of the proposed system are Confidentiality (Confidential information is any information that is not generally known and is conveyed or made available to the receiving party in a manner that does not allow for public disclosure.) and is Fast and Convenient.

IV. METHODOLOGY

In this proposed model, the first step is to gather the information or data from the open source; this will be used to train the models. Then the data need to be pre-processed according to the models; it helps to increase the model's accuracy and better information about the data. In the next step, Data from the columns is prioritized to choose which features to use; by this, the time used on many columns is reduced. To get the result, model building for the data set is an important step. Based on the dataset, build the model for classification and regression. Lastly, the user views the generated results from the model. *A) Support-Vector Machine (SVM)*

Support-vector machines (SVMs) are supervised learning models in the field of machine learning that use learning algorithms to analyse data for regression and classification. SVMs are among the most trustworthy methods of prediction since they are dependent on statistical learning frameworks. An SVM is a nonprobabilistic binary linear classifier that is taught to make predictions based on a set of training examples annotated with binary labels. When used for tasks like regression, classification, and outlier identification, SVM generates a hyperplane or group of hyperplanes in a high- or infinitedimensional space.

TABLE 1.
TESTCASES

S.NO	TEST CASES	INPUT	OUTPUT
1	Read the dataset	Dataset path	Dataset fetched successfully
2	Verifying the features and generates result	Input features as input, for Security Level classification for data image encryption	Output is classified as different Security Level
3	Verifying the features and generates result	Input features as input, for Security Level prediction for Lorenz image encryption	Model successfully predicted security level
4	Verifying the features and generates result	Input features as input, for Security Level prediction for no encryption	Model successfully predicted security level
5	Verifying the features and generates result	Input features as input, for Security Level prediction for rubix encryption	Model successfully predicted security level
6	Generating encryption and generates decrypted image	Input features as input, for Security Level prediction for logistic map	Model successfully encrypted and decrypted the image
7	Generating encryption and generates decrypted image	Input image as input, for Security Level prediction for logistic map	Model successfully encrypted and decrypted the image
8	Generating encryption and generates decrypted image	Input image as input, for Security Level prediction for rubix encryption	Model successfully encrypted and decrypted the image
9	Generating encryption and generates decrypted image	Input image as input, for Security Level prediction for data image encryption	Model successfully encrypted and decrypted the image

By describing the mappings in terms of a kernel function, SVM techniques guarantee that the dot products of pairs of input data vectors may be calculated in terms of the original space variables. *B) Xgboost (XGB)*

XGBoost stands for eXtreme Gradient Boosting. This algorithm is widely used nowadays as it was more scalable and very easy to improve speed and performance. Some of the reasons why this algorithm is greatly used are gradient tree boosting and shrinkage and column subsampling.

C) Random Forest

Random Forest has supervised learning models in the field of machine learning that use learning algorithms to analyse data for regression and classification [13]. It is greatly known and used for classification problems as it can perform well with a dataset which have continuous variables.

D) DNA Encoding

DNA computing is a kind of computing that, rather than using silicon-based computer technology, relies on genetic material (DNA), biochemistry, and molecular biology. Four bases make up a DNA sequence in a single strand. The hypothesis of DNA coding proposes that DNA sequences may be used to encode information. Thus, the four bases are expressed as binary integers [12]. Since DNA bases are related in a complementary way, only eight of the possible twenty-four coding permutations meet the notion of complementary base pairing.

E) Logistic map

Classical examples of how complicated, chaotic behaviour may arise from relatively basic nonlinear dynamical equations [7] include the logistic map, and a degree-2 polynomial (or recurrence relation). Reproduction, where the population expands at a rate proportional to the present population, may be described by this nonlinear difference equation when the population is small. As the current population is subtracted from the theoretical "carrying capacity" of the ecosystem, the growth rate will decrease proportionally. However, given certain initial conditions and settings, the logistic map has a problem with negative population numbers. Similar chaotic dynamics may be seen in the older Ricker model, but this problem is not present in it [8].

F) Rubik's Cube Image Encryption

This technique uses the same idea as a Rubik's cube to move about the pixels in a picture. Encryption is achieved by applying the XOR operator to the out-of place columns and rows of the image using a secret key. For images with asymmetrical rows and columns, the same key is assigned in the opposite direction. The suggested method was evaluated experimentally with thorough numerical analysis, and it is secure [10] against several assaults, including statistical and differential attacks (visual testing). In addition, studies evaluating performance show that the proposed picture encryption technique is safe. Encryption and decryption can be performed quickly, making it a good choice for real-time Internet encryption and transmission [11].

G) Lorenz Image Encryption

The Lorenz equation is just a model for fluid convection in the atmosphere driven by temperature differences. Since the attractor has two wings like a butterfly, it is thought to be responsible for the "butterfly effect" and is thus classified as a classically chaotic system. Chaotic control, Chaos theory, dynamic system [9] modeling, and synchronization have all devoted considerable time and energy to studying the phenomena of synchronization. Consider that the Lorenz chaotic equation is a threedimensional dynamical system. When the starting circumstances are varied, the equation system exhibits a chaotic pattern of behaviour [14]. The Lorenz system's chaotic behaviour is much more nuanced than that of any 1D or 2D chaotic system. To encode pictures, the Lorenz equation is used.

H) Modules

There are two modules; they are User and System. User consists of Data Gathering, Pre-Processing, Feature Engineering, Model Building and View Results. The system consists of Model Checking and Generate Results.

The process starts with collecting the dataset from valid resources and inputting the dataset into the system. Now the dataset should be trained and tested so that it learns the process and can be implemented in the further steps. Next, the required machine learning models and encryption algorithms are executed on the dataset. Now the image is taken as the input, and after executing the models, the output is the encrypted image, and it also displays whether the algorithm is strong or weak based on the accuracy.

After opening the album icon that is displayed on the home page, then the choose file option is given. By clicking on the choose file we can select the image and then select the below displayed ML algorithm and one encryption algorithm of own choice and click on upload. After successful uploading the encrypted image gets displayed and accuracy score will be displayed.

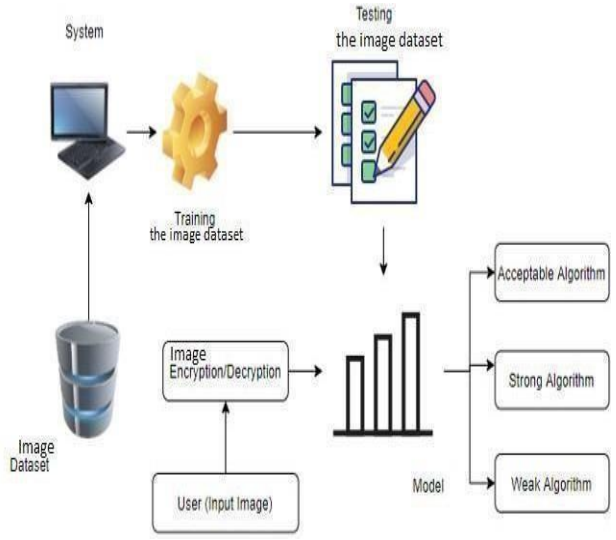


Fig. 1 System Architecture

	Entropy	Energy	Contrast	Correlatio	Homogen	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

Fig. 2 Data Set

In the dataset, the total number of entries is 2939, and the parameters used are Security, Peak signal to noise ratio, Homogeneity, Correlation, Contrast, Energy, Entropy, Mean Square Error, Peak signal to noise ratio, Security.

TABLE 2
RESULT

Input	Output	Result
Input features	Tested for different features given by user on the different model.	Success
Security Levels classification	Tested for different input features given by the user on different features from the models are created using the different algorithms and data.	Success
Security Levels Prediction	Security level prediction will be performed using the different models build from the algorithms.	Success

The figure 3 shows the basic input page where the image to be chosen and uploaded to get the encrypted image using machine learning models and encryption algorithms.

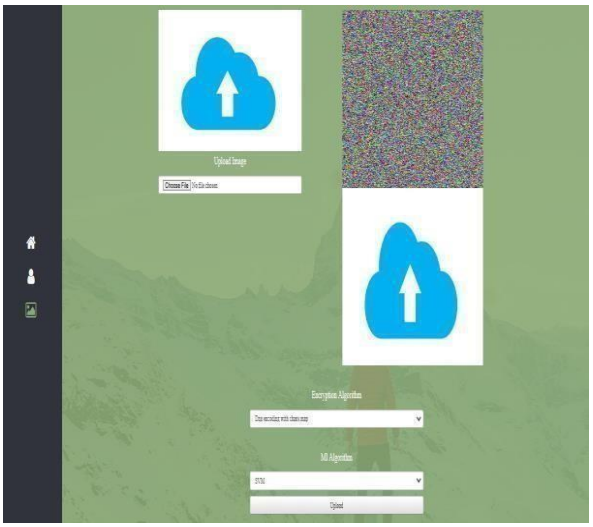


Fig. 3. Before Input

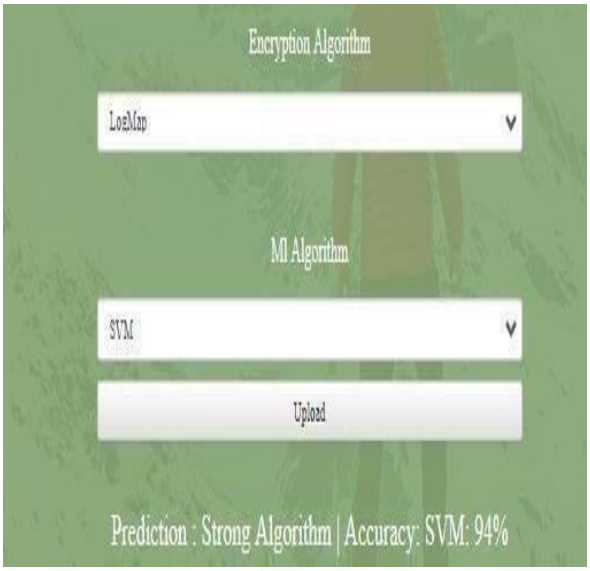


Fig. 4. Selection of Encryption and ML algorithms

The figure 4 shows the menu where encryption algorithms and machine learning algorithms can be selected.

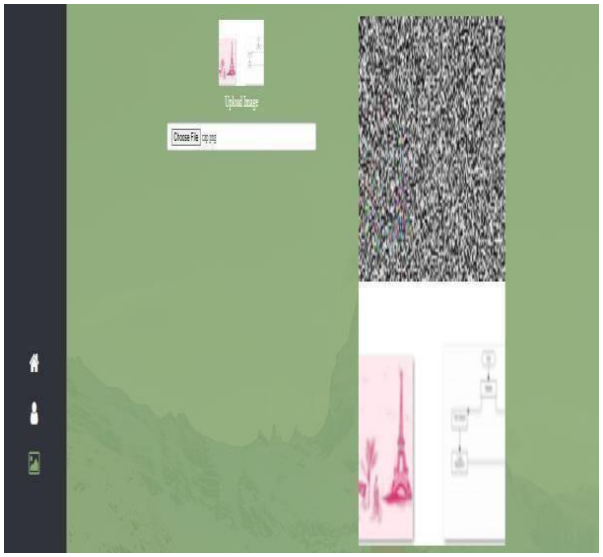


Fig. 5. Output

The figure 5 shows the output page where the encrypted image is displayed.

V. RESULTS AND DISCUSSION

As the main motive of the research on this project is to get the strongest encryption algorithm to secure the security levels of the given Image, tried using various encryption algorithms are used, such as DNA encoding, Log map, Rubix, Lorenz, and no encryption algorithm, to check every possibility for getting the most accurate algorithm for the process. Machine Learning models are also used such as SVM, XG Boost, and Random Forest. As a result, the accuracy and to state whether the algorithm is strong, weak, or acceptable is executed.

VI. CONCLUSION

It provides a mechanism for efficiently and correctly determining the degree of security offered by different encryption techniques. As a first step, it has amassed a dataset and incorporated as characteristics the security parameters shared by different encryption methods. In order to compile a data set, it classifies the possible ranges of feature values into 3 categories: "strong," "acceptable," and "weak," which correspond to three distinct degrees of security. The proposed model is then used to compare the relative safety of different forms of encryption. In addition, it can identify the security of various encryption methods automatically by calculating their individual statistical values. This takes a long time to complete using conventional testing techniques, but with the suggested approach, testing may be done in a matter of seconds. Finally, this suggested model is evaluated using various experiments to assess its performance, and it is shown to

provide 94% accurate predictions at much higher speeds than existing models.

REFERENCES

[1] Automated detection and classification of cryptographic algorithms in binary programs through machine learning by Diane Duros Hosfelt(2015).

[2] Applications in Security and Evasions in Machine Learning: A Survey by Ramani Sagar 1, *, Rutvij Jhaveri 2 and Carlos Borrego 3(2020).

[3] Machine learning approaches to IoT security: A systematic literature review By Rasheed Ahmada, Izzat Alsmadi (2021).

[4] Secure, privacy-preserving and federated machine learning in medical imaging By Georgios A. Kaissis, Marcus R. Makowski, Daniel Rückert & Rickmer F. Braren (2018)

[5] Machine Learning and Cryptographic Algorithms –Analysis and Design in Ransomware and Vulnerabilities Detection by Nandkumar Niture (2020).

[6] Prediction of DDoS Attacks using Machine Learning and Deep Learning Algorithms by Saritha, B. RamaSubba Reddy, A Suresh Babu (2017).

[7] I. Hussain, A. Anees, A. H. Alkhaldi, M. Aslam, N. Siddiqui, and R. Ahmed, ‘‘Image encryption based on Chebyshev chaotic map and S8 Sboxes, (2019).

[8] A. Anees, I. Hussain, A. Algarni, and M. Aslam, ‘‘A robust watermarking scheme for online multimedia copyright protection using new chaotic map,’’ (2018).

[9] A. Shafique and J. Ahmed, ‘‘Dynamic substitutionbased encryption algorithm for highly correlated data, “(2020).

[10] F. Ahmed, A. Anees, V. U. Abbas, and M. Y. Siyal, ‘‘A noisy channel tolerant image encryption scheme,’’ (2014).

[11] L. Lee and W.-H. Tsai, ‘‘A new secure image transmission technique via secret-fragment- visible mosaic images by nearly reversible color trans formations,’’ (2014).

[12] M. Khalili and D. Asatryan, ‘‘Colour spaces effects on improved discrete wavelet transformbased digital image watermarking using Arnold trans form map,’’ (2013)

[13] Felix, A. Yovan and Thankappan Sasipraba. “Flood Detection Using Gradient Boost Machine Learning Approach.” *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)* (2019): 779-783.

[14] Felix, A. Yovan, and T. Sasipraba. "Decision support system for flood risk assessment and public sector performance management of emergency scenarios." International Journal of Public Sector Performance Management 8, no. 3 (2021): 219-229