ML in CryptAnalysis: Decrypting cipher text using Machine Learning Paradigm

Sehban Fazili MT21143

Anam Fatima PhD21016

Abu Osama Siddiqui MT22006

Abstract

In information security, the primary concern is protecting data against unauthorised access. Cryptography provides techniques for securing data such that only the sender and intended receiver can view its contents. Cryptanalytic attacks aim at deciphering this hidden information by guessing the plaintext or the key. In classical language modeling, deciphering historical ciphers such as substitution ciper is a challenging task. The proposed work attempts to optimize the decipher key-searching by making use of metaheuristic tool of Genetic Algorithm (GA) to perform cryptanalysis. A character-level language model that gives real words and phrases a higher log-likelihood while giving false words and sentences a low rating, has been used as the fitness function for GA.

1 Introduction

Information security is one of the most critical and challenging areas, given in to the vast amount of digital information accessible. Encryption of data while being stored or transmitted is the most widely accepted way to keep information secure. Cryptography ensures secure communication such that only the intended sender and receiver can access and process information.
The four essential goals of cryptography are: confidentiality, identification and authentication, integrity, and confidentiality. Most of the current cryptographic techniques aim to provide all of the aforementioned features or in combination. Broadly categorized, cryptographic algorithms are of two types: symmetric encryption algorithms using the same key to encrypt and decrypt a message, and asymmetric which use different keys for encryption and decryption. Most of the historical cryptographic systems are based on symmetric key cryptography providing only information confidentiality. One of the earliest techniques was substitution ciphers going as back to Julius Caesar (100 B.C. to 44 B.C.). Also known as Caesar Cipher, it is one of the simplest substitution cipher schemes where each letter in plaintext is replaced by another character to form ciphertext. Some other historical techniques include transposition cipher, polyalphabetic substitution, and permutations cipher. Modern symmetric key cryptography includes block cipher schemes such as DES and AES algorithms and stream cipher techniques.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

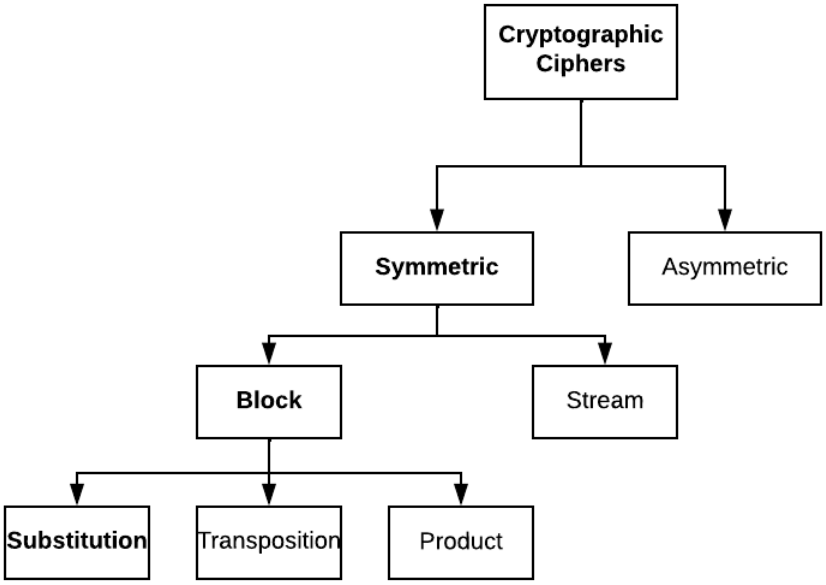


Figure 1: Cryptographic cipher classification (8)

Cryptanalysis is the study of methods/ techniques used to convert ciphertext to plain text without access to secret key used for encryption/ decryption. Also known as codebreaking or cracking code the unauthorized person attempts to decipher the information. The aim of the attacker attempting to get unauthorized access to information(passive attack) or, worse, alter/ modify the information (active attack). Such attackers use various cryptanalysis techniques for understanding ciphertext, ciphers, and cryptosystems to get access to secret information. One of the classical attacks is brute force attack or exhaustive search which relies on guessing all possible combinations of a targeted text or character combinations to find the key that yields most probable text. However, trying all possible combinations of alphabets for the given key or text in exhaustive search is not feasible considering time and efficiency. Hence to save time and effort, it became a necessity to look for some other techniques to find optimal solution for choosing permutation of 26-character alphabet used to decipher ciphertext.

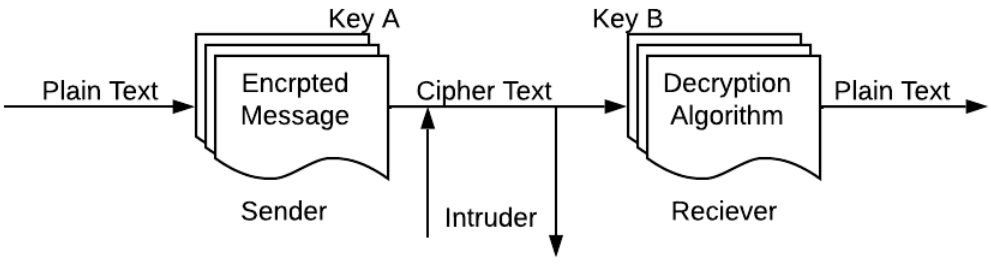


Figure 2: Flow of the project

Genetic Algorithms(GA) have been proposed in number of works as a generic meta heuristic technique for solving problems having no satisfactory or obvious solutions. Based on biological evoution and natural selection, it can be used to solve constrained/ unconstrained optimization tasks. It starts with defining a fitness function and creating a DNA string for genetic representation, proceeding by initialization of population of DNA string as proposed solutions and then improving it through iterative application of genetic operations as mutation, crossover, inversion and selection operators.

Generation process of offspring as probable solutions, is repeated until a termination condition such as minimum criteria solution is met.
In the proposed work, decipher of encrypted text is considered as an optimization task where the objective function is likelihood of decrypted text where model parameters are assumed to be the dictionary used to map encrypted letter to plain letter. Since, the objective function is not directly differentiable with respect to model parameters, GA has been proposed as intermediate solution to represent values of map and log likelihood of guessing the encrypted text as the fitness function. It proposes an end-to-end classical language model utilizing GA as optimization technique to find possible combinations of real english encrypted text, cutting down on time and efforts wasted by brute force techniques. GA uses a population of DNA strings representing values of the substitution cipher mapping, use it to decode message and compute the log likelihood of decoded message as probable original text. It iteratively mutates the DNA string pool, evaluating its fitness to decrypt the message. By the end of the loop, the algorithm will converge to generate the most likeable original message, assuming that real english sentences have higher log-likelihood than random letters. The language model can be extended to multi-lingual setup as well for breaking substitution ciphers.

2 Related Work

The term "cryptanalysis," which comes from the Greek words "kryptós," which means "hidden," and "analýein," which means "to loosen," refers to the art and science of breaking ciphertext into its equivalent plaintext without having access to the secret key. The first people to make a substantial contribution to cryptanalysis were Arabs. An Arabic author named Qalqashandi described a method for cracking ciphers early in the 15th century (1) by using the average frequency of each letter in the language. Several optimization techniques have shown promise for automated cryptanalysis of classical ciphers over the past years. Peleg and Rosenfeld (10) made one of the original proposals. They used a probabilistic labeling problem to represent the challenge of cracking substitution ciphers. Probabilities representing plaintext alphabets were allocated to each coded alphabet, and they were updated using the combined letters. They were able to decipher the cipher by repeatedly employing this strategy. To crack simple substitution ciphers using hand-coded heuristics, Carrol and Martin (4) created an expert system technique. Spillman (12) presented a genetic algorithm strategy to crack a substitution cipher for the first time in 1993. In order to find the key for a straightforward substitution cipher, he has investigated the potential of using a random type search. In accordance with the limitations imposed by the encryption symbols, David Oranchak (9) suggested a dictionary-based approach utilizing a genetic algorithm that encodes answers as plain text word placements. Omran, AL-Khalid and Al- Saady (8) also used genetic algorithm to break a mono-alphabetic substitution cipher.
Nada Aldarrab (3) et al. solve the decipherment problem for 1:1 substitution cipher by proposing a multilingual model where the only input available is encrypted text. With the key, encryption technique, and plaintext all unknown, the aim is to recover the plain text using a sequence-to-sequence model, viewing it as a language translation task trained on multilingual data. They make use of the character frequency analysis technique, assuming that the frequency distribution of characters remains the same in any sample drawn from a given language. The ciphertext is encoded using character-wise frequency rank, trained on attention-based encoder-decoder Transformer model for the character-based character-level neural machine translation (NMT).
In (6), Kambhatla et al. proposed a Neural Language Model (LM) based beam search algorithm for the decipherment of substitution ciphers, reducing error rates significantly. For the given task, they exploited pre-trained neural LMs using a multiplicative LSTM-based byte (character) level neural LM. A beam search technique is used to identify argmax for which the probability of the deciphered text is maximized, incrementally ranking the most likely substitutions based on the language model scores.
To perform cryptanalyse for Mono-alphabetic Substitution Cipher, (7) applied Genetic Algorithm(GA) and different combinations of its genetic operator to find the most efficient solution. For a given encrypted text, it uses a random set of keys as initial population to decrypt the given encrypted text using all keys in population. It applies various operators of Genetic algorithm such as fitness function, cross-over and mutation, iteratively to replace intial population with new generation till stopping criteria is met.
GAs are typically used to generate population(s) which can be used as feasible solutions for a given problem with reduced processing time. Applied in cryptanalysis as reduced time key space searching

tool in (2), to decipher ciphertext generated by Hill cipher technique where each plaintext represented as a vector of integer values is encrypted using a single multiplication by a square key matrix. Brute force way cannot be used to find correct key from all possible keys is not feasible solution, and hence generic algorithm was used to make a guess of the optimal key for saveing time and effort.
Bradley Hauer and Grzegorz Kondrak in (5) proposed decipherment process as a unsupervised problem, providing three methods to determine the original language of a text encrypted using a monoalphabetic substitution cipher. The methods are based on relative character frequencies, patterns of repeated symbols within words and outcome of a trial decipherment with the best technique achieving 97 percent accuracy over 380 languages.
Sujith Ravi and Kevin Knight in (11) proposed n-gram based model based on Shanon's theory of uncertainty for solving substitution ciphers. Their main contribution being that instead of using heuristic methods or expectation-maximization (EM) based methods, they proposed an exact letter-substitution decipherment method which ensures that ensures that no key is missed, and - may be implemented using standard integer programming solvers.

3 Dataset

We primarily extract/download text files from Project Gutenberg that are in the English language. Project Gutenberg is an online library of free eBooks. We use Herman Melville's Moby-Dick, also known as THE WHALE, from the Project Gutenberg eBook collection. The ebook contains 136 chapters and 216055 tokens/words.

4 Methodology

The steps involved are as follows:

1. Generating a random Substitution Cipher.
2. Read the corpus/dataset, creating a character-level language model.
3. Train the language model on the corpus.
4. Creating Encoding and Decoding functions.
5. Run an evolutionary/genetic algorithm to decode the message.
6. Results, compare the output to the original message.

The above-written steps can be summarized in three main topics as follows:

4.1 Substitution Cipher

Substitution Cipher is a type of symmetric cryptography cipher. There are two types of substitution cipher (Mono alphabetic and Poly alphabetic). In substitution ciphers, the position of the original string and its replacement value corresponds precisely to the plain and ciphertext, but the value of the character or character string is changed when the plaintext is converted into the ciphertext. In substitution ciphers, the plaintext is encrypted by replacing each letter or symbol with a different symbol as instructed by the key. In a substitution cipher, each letter of the plaintext is changed to a different letter, symbol, or number; the reverse substitution is required for decryption. The pseudocode for the generation of substitution we used is as follows:

1. Create two lists that contain lowercase letters and randomly shuffle the second list.
2. Initializing an empty dictionary
3. Map the to lists in the dictionary where the key is from List1 and the value is from List2.

Example of a substitution Cipher:
Text Characters:
'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'
Corresponding Key Cipher:
'n', 'a', 'b', 'k', 't', 's', 'o', 'w', 'f', 'z', 'e', 'p', 'r', 'g', 'd', 'q', 'u', 'c', 'j', 'm', 'h', 'y', 'i', 'v', 'x', 'l'

4.2 Language Models

We have developed a character-level language model that gives actual words and phrases a high likelihood rating while giving false words and sentences a low rating. Finding the translation/decryption that provides the greatest likelihood is what we're after. Therefore, in order to determine the likelihood of phrases, we must first determine the likelihood of a single word.
The probability formulas are as follows:
The probability of 2 lettered word is given by

$$P(AB) = P(B \mid A) * P(A)$$

where, P(B|A) is the conditional bigram probability and P(A) is the marginal unigram probability.
Now, if we have a 3 letters word then the probability is given by

$$P(ABC) = P(C \mid AB) * P(B \mid A) * P(A)$$

Now by Markov assumption the current state will depend on the previous state and not on any earlier states,

$$therefore, P(C \mid AB) = P(C \mid B)$$
$$P(ABC) = P(C \mid B) * P(B \mid A) * P(A)$$

So, the probability of words of any length t is given by

$$P(x_1, x_2, x_3, ..., x_t) = P(x_1) \prod_{t=2}^{T} P(x_t \mid x_{t-1})$$

Therefore the probability of a sentence is given by

$$P(w_1, w_2, w_3, ..., w_N) = \prod_{n=2}^{N} P(x_1^n) \prod_{t=2}^{t(n)} P(x_t^n / x_{t-1}^n)$$
$$where, w_n = x_1^n, x_2^n, x_3^n, ..., x_T^n$$

Limitation: There are occasionally unusual bigrams that don't show up in the training corpus but do in the message. As a result, the bigram's likelihood is reduced to zero, which also reduces the probability of the entire sentence. In order to resolve this issue, we must first add the numerator by 1 and the denominator by the total number of letters in the alphabet. Add-one smoothing is the name given to this process.
The pseudocode is as follows:

1. First, find the individual bigram/unigram probabilities using a large text corpus.

2. Calculate the probabilities.

We find that the probabilities we calculate are very small. To deal with this issue, we use log-likelihood.

$$P(x_1, x_2, x_3, ..., x_t) = logP(x_1) + \sum_{t=2}^{T} logP(x_t / x_{t-1})$$

4.3 Genetic Algorithm

The genetic algorithm, which is based on natural selection, the mechanism that propels biological evolution, is a technique for resolving both limited and unconstrained optimization issues. A population of unique solutions is repeatedly modified by the genetic algorithm. The genetic algorithm chooses members of the present population to serve as parents at each stage and employs them to produce the offspring that will make up the following generation. The population "evolves" toward the best option over the course of subsequent generations. The key algorithmic steps are shown in the flowchart given in the figure.

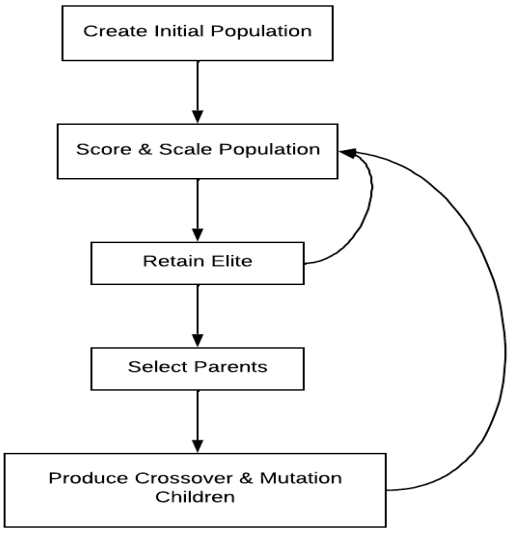


Figure 3: Flowchart of Genetic Algorithm

The pseudocode for the algorithm is as follows:

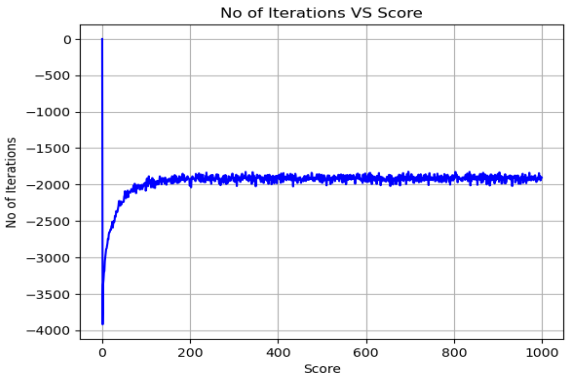
Algorithm 1 Genetic Algorithm

Ensure: $DNA_pool \leftarrow$get_random_dna(20)
  for i in range(20): **do**
    if $i > 0$ **then**
      $DNA_pool \leftarrow$create_offspring(DNA_pool, 3)
    Scores$\leftarrow$ [f(DNA) for DNA in DNA_pool]
$DNA_pool \leftarrow$sorted_DNA[:5]
end if $= 0$

5 Results

For several populations, the attack on a substitution cipher was applied. The plots below demonstrate the correlation between fitness (scores) and generations for populations of 20, 30, and 50. According to the statistics, the best fitness level is attained after 1000 generations. The best key found has all the correct letters and is ezjuwmrsqkyxhitplcavgndfbo. The Bleu score was utilized as the evaluation metric, and the results are shown in the table below.

	Population Size	Number of Iterations	Bleu Score
Plot 1	20	1000	0.8394
Plot 2	30	1000	0.8394
Plot 3	50	1000	0.1797
Plot 4	20	2000	0.8397



Plot 1

Plot 2

Plot 3

Plot 4

6 Analysis

The decrypted text is not always going to give us the best result. We might get some alphabets in the key wrong. The example of this analysis is as follows:

Original Message:
A long time ago a Man met a Satyr in the forest and succeeded in making friends with him. The two soon became the best of comrades, living together in the Man's hut. But one cold winter evening, as they were walking homeward, the Satyr saw the Man blow on his fingers. "Why do you do that?" asked the Satyr. "To warm my hands," the Man replied. When they reached home the Man prepared two bowls of porridge. These he placed steaming hot on the table, and the comrades sat down very cheerfully to enjoy the meal. But much to the Satyr's surprise, the Man began to blow into his bowl of porridge. "Why do you do that?" he asked. "To cool my porridge," replied the Man. The Satyr sprang hurriedly to his feet and made for the door. "Goodby," he said, "I've seen enough. A fellow that blows hot and cold in the same breath cannot be friends with me!"

Encrypted Message:
r lvno duiw rov r irn iwd r grdth un dbw evhwgd rns gqjjwwsws un irpuno ehuwnsg mudb bui dbw dmv gvvn fwjriw dbw fwgd ve jvihrswg lukuno dwowdbwh un dbw irn g bqd fqd vnw jvls mundwh wkwnuno rg dbwt mwhw mrlpuno bwiwmrhs dbw grdth grm dbw irn flvm vn bug eunowhg mbt sv tvq sv dbrd rgpus dbw grdth dv mrhi it brnsg dbw irn hwaluws mbwn dbwt hwrjbws bviw dbw irn ahwarhws dmv fvmlg ve avhhusow dbwgw bw alrjws gduriuno bvd vn dbw drflw rns dbw jvihrswg grd svmn kwht jbwwheqllt dv wnzvt dbw iwrl fqd i dv dbw grdth g gqhahugw dbw irn fworn dv flvm undv bug fvml ve avhhusow mbt sv tvq sv dbrd bw rgpus dv jvvl it avhhusow hwaluws dbw irn dbw grdth gharno bqhhuwslt dv bug ewwd rns irsw evh dbw svvh ovvsft bw grus u kw gwn wnvqob r ewllvm dbrd flvmg bvd rns jvls un dbw crew fhwrdb jrnnvd fw ehuwnsg mudb iw

Decrypted Message:
a long time ago a man met a catyr in the forect ans cuppeeses in making friensc with him the two coon bepame the bect of pomrasec living together in the man c hut but one pols winter evening ac they were walking homewars the catyr caw the man blow on hic fingerc why so you so that ackes the catyr to warm my hansc the man redlies when they reaphes home the man dredares two bowlc of dorrisge thece he dlapes cteaming hot on the table ans the pomrasec cat sown very pheerfully to enjoy the meal but muph to the catyr c curdrice the man began to blow into hic bowl of dorrisge why so you so that he ackes to pool my dorrisge redlies the man the catyr cdrang hurriesly to hic feet ans mase for the soor goosby he cais i ve ceen enough a fellow that blowc hot ans pols in the came breath pannot be friensc with me

Some of the words in the decrypted text are incorrect. The following incorrect predictions are made:
true: c, pred: p
true: d, pred: s
true: p, pred: d
true: q, pred: x
true: s, pred: c
true: x, pred: z
true: z, pred: q

7 Observations

1. The actual maximum likelihood might not be the correct response.
2. The likelihood of the true response may be lower.
3. The genetic algorithm is finding the maximum as intended.
4. Markov assumption, bi-grams are too restrictive.

8 Additional Study

We also considered creating our own dataset of plain text, cipher text and decrypted text using our model as well as extending the proposed work to multi-lingual setup.

9 Conclusion and Future Scope

In this project, a successful genetic algorithm attack on a given substitution cipher was carried out. The population size and the number of iterations, among other variables, were evaluated. The findings have shown that raising the population above 20 did not aid in locating the original key. Additionally, we found out that increasing the iterations is in fact, ineffective. We learned more about language models and optimization techniques, as a result. We hope to expand our strategy in future works, by utilizing deep learning models. Although the computation would be significantly costly, employing trigrams could potentially also serve the purpose

References

[1] AL-KADIT, I. A. ORIGINS OF CRYPTOLOGY: THE ARAB CONTRIBUTIONS. *Cryptologia 16*, 2 (Apr. 1992), 97–126.

[2] AL-KHALID, A. S., AND AL-KHFAGI, A. O. Cryptanalysis of a Hill cipher using genetic algorithm. In *2015 World Symposium on Computer Networks and Information Security (WSCNIS)* (Hammamet, Tunisia, Sept. 2015), IEEE, pp. 1–4.

[3] ALDARRAB, N., AND MAY, J. Can Sequence-to-Sequence Models Crack Substitution Ciphers? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (Online, 2021), Association for Computational Linguistics, pp. 7226–7235.

[4] CARROLL, J. M., AND MARTIN, S. The automated cryptanalysis of substitution ciphers. *Cryptologia 10*, 4 (1986), 193–209.

[5] HAUER, B., AND KONDRAK, G. Decoding Anagrammed Texts Written in an Unknown Language and Script. *Transactions of the Association for Computational Linguistics 4* (04 2016), 75–86.

[6] KAMBHATLA, N., MANSOURI BIGVAND, A., AND SARKAR, A. Decipherment of Substitution Ciphers with Neural Language Models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Brussels, Belgium, 2018), Association for Computational Linguistics, pp. 869–874.

[7] MUDGAL, P. K., PUROHIT, R., SHARMA, R., AND JANGIR, M. K. Application of Genetic Algorithm in Cryptanalysis of Mono-alphabetic Substitution Cipher. 6.

[8] OMRAN, S. S., AL-KHALID, A. S., AND AL-SAADY, D. M. Using genetic algorithm to break a mono - alphabetic substitution cipher. In *2010 IEEE Conference on Open Systems (ICOS 2010)* (2010), pp. 63–67.

[9] ORANCHAK, D. Evolutionary algorithm for decryption of monoalphabetic homophonic substitution ciphers encoded as constraint satisfaction problems. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation* (New York, NY, USA, 2008), GECCO '08, Association for Computing Machinery, p. 1717–1718.

[10] PELEG, S., AND ROSENFELD, A. Breaking substitution ciphers using a relaxation algorithm. *Communications of the ACM 22*, 11 (Nov. 1979), 598–605.

[11] RAVI, S., AND KNIGHT, K. Attacking Letter Substitution Ciphers with Integer Programming. *Cryptologia 33*, 4 (Sept. 2009), 321–334.

[12] SPILLMAN, R., JANSSEN, M., NELSON, B., AND KEPNER, M. Use of a genetic algorithm in the cryptanalysis of simple substitution ciphers. *Cryptologia 17*, 1 (1993), 31–44.