Journal of Scientific & Industrial Research
Vol. 83, January 2024, pp. 25-33
DOI: 10.56042/jsir.v83i1.5324

Cryptoanalysis of Simple Substitution Ciphers with Genetic Algorithms

Dragan Savić[1], Petar Milić[2]* & Borislav Mazinjanin[3]

[3]Singidunum University, Danijelova 32, 11000 Belgrade, Serbia

[2]University of Pristina — Kosovska Mitrovica, Faculty of Technical Sciences, Knjaza Miloša 7, 38220 Kosovska Mitrovica, Serbia

[3]Oxoftware, Bulevar Marsala Tolbuhina 44, 11070 Novi Beograd, Serbia

Received 02 September 2023; revised 17 October 2023; accepted 28 December 2023

Recent advances in the field of machine learning have once again raised the question whether computers can be trained to perform cryptanalytic tasks. In this paper, we identify the relationship between the machine learning and cryptanalysis with special attention on the genetic algorithms as a heuristic optimization method inspired by evolution processes in nature. We have indicated the consequences that new insights of machine learning may have on the reformulation of the practical criteria of secrecy in the synthesis of information security systems. Through the paper, we describe the approach based on genetic algorithms in order to confirm which machine learning algorithms are most suited for the purpose of the cryptanalysis and consequently to verify resistance of encryption algorithms to cryptanalysis.

Keywords: Analysis, Cryptanalytic tasks, Encryption, Genetic algorithms, Machine learning

Introduction

Cryptanalysis has been for ages considered a fascinating area that is placed between art and science. The reason for this lies in the fundamental limitations, as well as in the capacities of theoretical physics, computer science and mathematics. The complex game between cryptanalysts and cryptographers is played in a sort of the Bermuda Triangle. Is the secret of the cipher system created by people indeed unsolvable? Edgar Allan Poe had a strong opinion on the subject stating that enigmas created by human ingenuity could only be solved by human ingenuity.[1]

Claude Shannon, in two of his foundation works[2,3], gave the first mathematically precise formulation of the secret of the cipher system and how to measure it. Cryptography and cryptanalysis have since developed as two different scientific fields. In literature, many authors consider cryptography and cryptanalysis as subareas of the theory of information and coding.[3,4] A good overview of the most significant cryptanalysis techniques was given by Prajapat and Thakur.[5] They proved what has been known all along - it is far easier to design a cipher system than to decipher one. Behind cryptanalysis techniques are the complex mathematical models of statistical resolution and the analysis of an algebra structure over the finite fields, with the purpose of revealing of the encryption scheme. It is not uncommon for some artificial intelligence techniques to be used for cryptanalysis due to the reason that they cope well with large amount of data for analysis.[6]

Fundamentally, the method comes down to the logical analysis of a cipher transformation, i.e. its translation into an appropriate conjunctive normal form over the variables of interest. For example, variables can be secret internal keys of the analyzed cipher system.[7–10] A deeper foundation of the analogy is derived from more general Cook's result regarding equivalence of any program that is stopped on the Turing Machine and corresponding system of Boolean equations.[3] Why are the completely different methods of cryptanalysis being searched for? One of the possible answers can be found in famous Shannon's concept of the unicity distance.[2] In accordance with Shannon's concept, with a cipher longer than the unicity distance practically means that a cryptanalyst have a complete information on the secret key, i.e. a cryptanalytic attack can lead to a possible complete reconstruction of the secret key based solely on a cipher in the information theory sense.[3] The manifestation of a cipher system through cipher text completely reveals the confidentiality of an internal encryption key. The only problem with Shannon's result is that the general reconstruction procedure has not been established yet apart from the

*Author for Correspondence
E-mail: petar.milic@pr.ac.rs

attack of a complete search on the keys space. This is one of the strongest motives for the search of a general cryptanalytic attack. Hence, numerous resources of all possible attempts in this area do not come as a surprise.

In this paper, we will present the relationship between general cryptanalytic attack based on the reformulation of the cryptanalysis problem and the machine learning approaches especially genetic algorithms (GA). GAs are indispensable in cryptanalysis due to their efficiency in searching vast solution spaces, adaptability to evolving encryption methods, parallel processing capabilities, versatility in handling non-linear cryptographic problems and heuristic optimization. GAs provide a heuristic, customizable, and secure approach to decrypting messages and breaking encryption, making them essential tools for tackling complex cryptographic challenges. This paper offers assessment that relies on GAs to determine the machine learning algorithms that are best suited for the task of cryptanalysis. This, in turn, helps us assess the resilience of encryption algorithms to cryptanalysis. Therefore, we firstly begin with related work review, after which we proceed with description between various operation modes of a cipher system and a corresponding machine learning system. After that, we provide reader with the prerequisites for the efficient implementation of this class of attacks, as well as the characteristics which corresponding training system must possess. In this regard, the use of the latest concepts of machine learning known as deep learning[11] in the cryptanalysis of cipher systems will be analyzed with special attention on the GAs. In penultimate section, we describe experiments we performed in order to verify the previously described approach. At the end of the paper we give conclusion and future work remarks.

Related Work

The results that can be found in the literature over the past few years shows the shifting from the shallow to deep architecture of training systems.[11–13] The depth of a structural scheme is defined by the greatest number of functional elements founded along the random path of signal emission from input toward output. In accordance with this definition, the SVM classifiers have the depth of 2, while the multilayer perceptron with one hidden layer has the depth of 3.

The general approach is that the depth of the training system architecture is directly related to the number of internal representations arranged in a strict hierarchical order. For example, if we depict a multilayer perceptron with n hidden layers, then system outputs are formed on the basis of the signal (internal representation) emitted from n-1 layer. Signals emitted from n-1 layer are formed on the basis of the previous internal representation (signal) emitted from n-2 layer, and etc. The hierarchical relationship of the sets of signals according to layers is caused by a way in which a signal spreads from the input to output of the training system. It appears that learning of deep architectures is much harder than that of the shallow ones. Hence, if classic ways of learning shallow architectures are applied, poor results are obtained.

In his research on the influence of the successive layers on the self-learning, Becker[14] shows that using the criteria of self-association accuracy are considered to be the turning point. This phase of self-learning can be understood as pre-training, which is then followed by the final classic monitored learning. A great aggregate result of this way of learning can be explained by considering the self-learning single layers of deep architecture as a kind of coding of a priori information about the problem which is being solved. Furthermore, Lee *et al*.[15] confirms that machine learning classifiers can be trained to predict a cipher's security margin based on the cipher features such as the number of rounds, permutation pattern and truncated differences. The proposed methodology achieves an accuracy of 62% supporting thus the feasibility of proposed approach. This is in line with work of Ja *et al*.[16] who claims that predicting cipher's type depends on the modelling of the learning approach as well as managing training processes rather than spending the time in mathematical computation of the cipher.

It is known that many cryptographic methods and algorithms requires a large computational power in order to provide better results. Andonov *et al*.[17] in his study of machine learning application in cryptanalysis, shows that neural networks represents adequate approach for precise cipher text attack. By application of different neural networks with the outside error and the total accuracy as a measures of attack, they obtain approximately 97% of cipher text accurately decrypted. Also, neural networks represents feasible machine learning approach to find a key from known plain text — cipher text pairs, especially when they are applied to the lightweight

block ciphers.[18] They can help in efficient and automated testing for checking the safety of emerging incomplete rounds of lightweight block ciphers.

According to the aforementioned discussion regarding the application of deep architectures of learning systems in cryptanalysis we can draw the following conclusions:

- The deep architecture can be adjusted to a factual information flow of the cipher system that is under attack. In such a manner, the architecture itself codes out a priori information on how the encryption or decryption procedure works, depending on the particular purpose of the learned system.
- Pre-learning based on self-learning represents a major gain in cryptanalytic scenario since the unmarked samples (ciphers) are always available.
- Pre-learning can be conducted off-line for each cipher transformation and delivered in that form as a kind of a semi-product for additional learning adjusted to a specific cryptanalytic activity.

Scenarios of use of the Learning Systems — Identification and Cryptanalysis

Taran, Rezaifar and Voloshynovskiy have provided a systematic study of the machine learning applicability in cryptanalysis.[19] Their research was followed by numerous studies which theoretically investigated what was possible to learn and what was not within the cope of classes typical for cryptographic primitives. A series of experimentally oriented studies show that the application of these techniques in practical cryptanalysis scenarios still has modest results.[20–23] Errors in the design of experiments and the analysis of obtained results can be noticed in many studies. A lack of double cross-validation schemes[24–26] which guarantee the objectivity of a machine learning model selection and its training for testing and obtaining firm estimates of reconstruction errors of aimed variables in the future operation environment of a system (generalization error) is typical.

Identification of Cipher Transformations by Learning Systems

Identification of a cipher transformation, i.e. the restoration of an input-output transformation without knowing internal secret cryptographic key K is the aim of the cryptanalysis. As we can see from Fig. 1, the equivalence between an encryption system with an unknown internal secret key and a learning system which identifies mapping between input Open Text (OT) and output Cipher (C) can be established, while Fig. 2 depicts vice-versa process.

In this case, the required training sets are pairs {OT,C} and {C,OT}. If the identification of the cipher transformation is successful - which is measured by a small error of generalization (a small error of reconstructed mapping over the independent testing sets), in this way identified system will be able to emulate encryption and decryption, which cannot be distinguished from the legitimate encryption and decryption even if the internal secret cryptographic key is unknown.

On the other hand, as long as the internal key is not changed, the identified cipher transformation for decryption is also a cryptanalytic system that decrypt (reconstructs) messages based on the given cipher even though the cryptanalyst does not have the internal secret cryptographic key.

Two general issues remain open while using learning systems in this manner:

- The system type and architecture of a learning system,
- The size of a training set for learning system.

Use of the Training Systems in Cryptanalysis

The most important achievement in cryptanalysis is the so-called total break of a cipher system when entire internal secret cryptographic key can be obtained through the available observations. As illustrated on Fig. 3, an equivalent training system was given, which enables the reconstruction of mapping required for obtaining the internal key

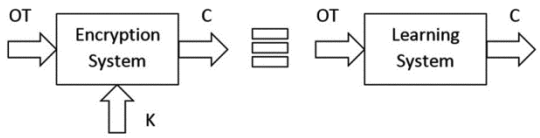

Fig. 1 — The use of a learning system in the encryption process

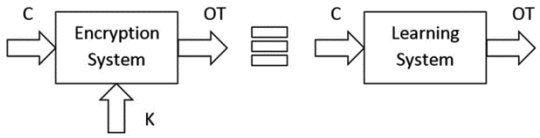

Fig. 2 — The use of a learning system in the decryption process

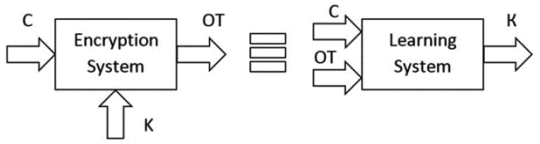

Fig. 3—A cryptanalytic use of a learning system in the process of decryption

through the observation of an open text and a cipher. Training sets required for training of this system are structured as {(C,OT),K} and can be obtained by simulating a cipher transformation.

This kind of training set formation is followed by a reliable piece of information on a cipher transformation which is a cipher algorithm, and by the information on the all details of the encryption algorithm. Information is publicly available for all standard encryption algorithms such as DES, 3DES, AES, RC4 which are predominantly used on the Internet and in mobile phones nowadays.

In Fig. 4 we provide a cryptanalytic use of the training systems when cryptanalysis refers to a partial reconstruction of the internal key. In this case, training sets have the structure of{(C,OT), (k_1,k_2,…,k_n }, if only n bits of the internal key is reconstructed, n≤|K|.

An expanded scenario of the cryptanalytic attack can be supplemented with information from so-called side channels as we shown in Fig. 5. Side channels are information channels that appear when cryptosystems operate in real time conditions, which system designers have not foreseen or could not disable and which can provide cryptanalyst with useful information. For example, a time diagram of power consumption of the processor on which cryptographic algorithm is applied, can be highly correlated with bits of the internal secret key or some of its correlates.

Additionally, time diagrams presenting the duration of certain commands execution on the processor can carry significant data on operators over which the given commands are executed, etc. In this case, training sets have the structure of {(C,OT,S),K},

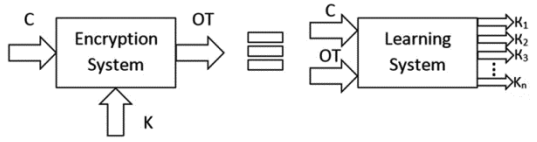

Fig. 4—Cryptanalytic use of learning system during decryption

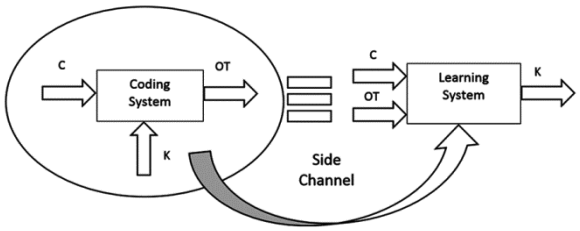

Fig. 5—Influence of side channel signals between cryptanalytic attack and corresponding learning system

where S are the corresponding signals of side channels. A clear attack based solely on signals emitted by side channels can easily be determined from the more general case by discarding input (C,OT) and retaining only input S. As far as outputs in both scenarios are concerned, they can be of either whole or partial value of the internal key.

Automatic Cryptanalysis of Simple Substitution Ciphers with GAs

GAs are a heuristic optimization method inspired by evolution processes in nature.[27] The basic idea can be summarized in the following. In each population, some individuals are better adapted to the environment than others, and so it is expected that they will reproduce more successfully. Since offspring inherits a combination of parental genes (while it is also able to mutate), every next generation is anticipated to be better adapted to the environment than the previous one. GAs offer a comprehensive set of qualities that make them exceptionally well-suited for cryptanalysis. Keeping in mind their global search capabilities, adaptability to evolving security measures, parallel processing prowess, ability to handle non-linearity, heuristic nature, customization options, and robust security measures collectively position GAs as a vital and effective tool for deciphering encrypted information and solving cryptanalysis challenges.

The basic algorithm can be represented by the following steps:

1. An initial generation containing n chromosomes with the l length of elements is created, where each chromosome represents a potential solution to the given problem. In the next steps, this generation is called the next generation.
2. The fitness is calculated for each chromosome x in the current generation f(x). If there is chromosome xˆ in the generation whose adaptability level meets the criteria for solving the problem, the execution of the algorithm stops and xˆ is considered to be the solution to the given problem. If not, step 3 is executed.
3. Steps a), b), c) are repeated until generations of n offspring (i.e. new solutions) are generated:
   (a) Two chromosomes are selected from the current generation, where better adapted chromosomes are being chosen with greater probability.
   (b) With a given value p_c, the crossover of selected chromosomes is performed, generating the set of offspring.

(c) Each offspring individually mutates with certain probability. Then the offspring is placed within a new generation.

4. A new generation of chromosomes becomes the current generation and the execution of the algorithm returns to step 2.

Generally, defining the structure of chromosomes and the processes of their crossover and mutation depend on the problem being modelled, with the restriction that the crossover and mutation should not generate a chromosome whose structure deviates from the acquired chromosome structure. A GA for the observed problem of automatic cryptanalysis of a simple substitution cipher is described below. Here we would like to mention that we have used adaptive GAs, which apply adaptive mechanisms to dynamically adjust their parameters during the optimization process, allowing them to respond to changes in the problem landscape.

Use of the Training Systems in Cryptanalysis

In the described approach, the cryptanalysis of the simple substitution cipher requires permutation π of alphabet Σ which maps the cipher text s_1 s_2…s_k into the plaintext. Therefore, every permutation of the alphabet will be considered to be a potential solution to the problem, and the chromosome will be defined as a string containing a permutation π of alphabet Σ.

Example 1. Eq. (1) - (4) contain four alphabet maps. Corresponding chromosomes are given below.

$$\pi_1 = \begin{pmatrix} a\,b\,c\,d\,e\,f\,g\,h\,i\,j\,k\,l\,m\,n\,o\,p\,q\,r\,s\,t\,u\,v\,w\,x\,y\,z \\ i\,k\,z\,x\,h\,a\,v\,p\,l\,j\,s\,y\,u\,c\,t\,e\,n\,q\,r\,g\,d\,m\,f\,w\,b\,o \end{pmatrix} \quad \dots (1)$$

$$\pi_2 = \begin{pmatrix} a\,b\,c\,d\,e\,f\,g\,h\,i\,j\,k\,l\,m\,n\,o\,p\,q\,r\,s\,t\,u\,v\,w\,x\,y\,z \\ p\,r\,u\,l\,z\,b\,e\,x\,v\,k\,s\,g\,i\,n\,m\,a\,q\,d\,f\,t\,w\,o\,y\,c\,h\,j \end{pmatrix} \quad \dots (2)$$

$$\pi_3 = \begin{pmatrix} a\,b\,c\,d\,e\,f\,g\,h\,i\,j\,k\,l\,m\,n\,o\,p\,q\,r\,s\,t\,u\,v\,w\,x\,y\,z \\ f\,z\,v\,d\,k\,e\,b\,j\,w\,q\,a\,c\,y\,p\,h\,r\,n\,m\,t\,x\,i\,u\,o\,g\,l\,s \end{pmatrix} \quad \dots (3)$$

$$\pi_4 = \begin{pmatrix} a\,b\,c\,d\,e\,f\,g\,h\,i\,j\,k\,l\,m\,n\,o\,p\,q\,r\,s\,t\,u\,v\,w\,x\,y\,z \\ e\,o\,b\,x\,d\,u\,n\,p\,q\,c\,h\,t\,z\,v\,k\,a\,l\,i\,w\,g\,y\,f\,s\,m\,j\,r \end{pmatrix} \quad \dots (4)$$

Chromosome adaptability measurement π can be estimated as the probability of obtaining an open text by mapping the given code s_1 s_2…s_k with the chromosome, i.e.,

$$f(\pi) = P\big(\pi(s_1)\pi(s_1)\dots\pi(s_k)\big) \quad \dots (5)$$

where, π denotes the chromosome, f(π) is chromosome π adaptability measurement, and π(s_1) π(s_2)…π(s_k) is the plaintext obtained by mapping cipher text s_1 s_2…s_k with chromosome π.

If the Eq. (6) is applied,

$$P(o_1 o_2 \dots o_k) \approx log\frac{C(o_1)}{\sum_{o_x \in ?} C(o_x)} + \sum_{i=2}^{k} log\frac{C(o_{i-1}o_i)+1}{\sum_{o_x \in ?} C(o_{i-1}o_x)+|\sum|} \quad \dots (6)$$

the result is:

$$f(\pi) = P\big(\pi(s_1)\pi(s_1)\dots\pi(s_k)\big) \approx log\frac{C(\pi(S_1))}{\sum_{S_x \in ?} C(\pi(S_x))} + \sum_{i=2}^{k} log\frac{C(\pi(S_{i-1})\pi(S_i))+1}{\sum_{\pi(S_x) \in ?} C(\pi(S_{x-1})\pi(S_x))+|\sum|} \quad \dots (7)$$

Example 2. Let's presume that we have the following cipher text: "vjururhapwhzcip". According to Eq. (7), chromosome fitness (1) — (4) are:

$$f(\pi_1) = P\big(\pi_1(vjururhapwhzcip)\big) = P(mjdqdqpiefpozle) = -70.74802657379654 \quad \dots (8)$$

$$f(\pi_2) = P\big(\pi_2(vjururhapwhzcip)\big) = P(okwdwdxpayxjuva) = -63.8493424363967 \quad \dots (9)$$

$$f(\pi_3) = P\big(\pi_3(jvjururhapwhzcip)\big) = P(uqimimjfrojsvwr) = -70.8061410964481 \quad \dots (10)$$

$$f(\pi_4) = P\big(\pi_4(vjururhapwhzcip)\big) = P(fcyiyipeasprbqa) = -60{,}64730024825707 \quad \dots (11)$$

Equation (7) includes Laplace's correction of the estimation of the maximum expectance of bigrams. In examples such as the previous one, in which the input data does not contain bigrams outside the dictionary, a version of the equation without Laplace's correction can be applied:

$$f(\pi) = P\big(\pi(s_1)\pi(s_1)\dots\pi(s_k)\big) \approx log\frac{C(\pi(S_1))}{\sum_{S_x \in ?} C(\pi(S_x))} + \sum_{i=2}^{k} log\frac{C(\pi(S_{i-1})\pi(S_i))}{\sum_{\pi(S_x) \in ?} C(\pi(S_{x-1})\pi(S_x))} \quad \dots (12)$$

Using this equation, the estimated adaptation of the observed chromosomes would change, but this will not have a conceptual impact on the exposition that follows. Moreover, in order to increase the accuracy of chromosome adaptability estimation (i.e., the probability of the occurrence of an open text), it would be more appropriate to use trigram characters[28] or word-level n-grams[29] in practical applications. Choosing an optimal approach to assess the probability of an open text depends on the particular problem specification. It should be remembered that, although the selected approach does not conceptually change the stated idea of automatic cryptanalysis of a simple substitution code, it can affect the algorithm effectiveness. In the following, it will be assumed that a suitable approach is chosen.

Chromosome Selection

Within a generation, chromosomes with larger fitness are selected with higher probability than the less fitted chromosomes. There are several ways to

select chromosomes, and two will be presented here — rank selection and tournament selection.

When selecting chromosomes by rank, it is assumed that a certain rank is assigned to each chromosome. If there is chromosome n in each generation and every chromosome in the generation is arranged by no decreasing fitness, starting with the least adapted chromosome to the best adapted chromosome, the first chromosome in this order gets rank 1, the second chromosome gets a rank equal to 2, etc., to the last chromosome that gets a rank equal to n. The probability of selecting a chromosome π from the generation $\wp$ s is defined with the following equation:

Using this equation, the estimated adaptation of the observed chromosomes would change, but this will not have a conceptual impact on the exposition that follows. Moreover, in order to increase the accuracy of chromosome adaptability estimation (i.e., the probability of the occurrence of an open text), it would be more appropriate to use trigram characters[28] or word-level n-grams[29] in practical applications. Choosing an optimal approach to assess the probability of an open text depends on the particular problem specification. It should be remembered that, although the selected approach does not conceptually change the stated idea of automatic cryptanalysis of a simple substitution code, it can affect the algorithm effectiveness. In the following, it will be assumed that a suitable approach is chosen:

$$P_r(\pi) = \frac{r(\pi)}{\sum_{\pi_x \in ?} r(\pi_x)} = \frac{r(\pi)}{\sum_{i=0}^{n} i} = \frac{r(\pi)}{\frac{n(n+1)}{2}} \quad \ldots (13)$$

where, r(π) is the rank of chromosome π and n is number of chromosomes in the generation.

The number of chromosomes in the generation is constant, so the chromosome selection probability is proportional to its rank. In other words, higher ranked chromosomes (i.e. with greater adaptability) are chosen with greater probability, which corresponds to the basic idea of the GA.

The probability distribution of chromosome selection is not uniform, so a brief review of the software implementation of the rank selection will be briefly discussed. Programming languages usually contain the functionality of pseudo-random numbering with uniform distribution. Having the principle of generality in mind, may it be assumed that function rand(n) returns the random integer from the set{1,2,….,n}, while each number is being chosen with the same probability(1/n). In the following, it will be explained how this function can be used to model non-uniform probability distribution defined by the chromosome rank.

Another way to select chromosomes is tournament selection which is performed through the following steps:

- Two chromosomes, π_i and π_j, are randomly selected from the current generation.
- The chromosome of greater adaptability is chosen between chromosomes π_i and π_j with probability p_t or with probability (1-p_t); the chromosome of less adaptability is chosen, where p_t is the pre-set parameter for which it is valid: 0,5<p_t≤1.

The probability of tournament selection of chromosome π from the generation that contains n chromosomes can be represented by the conditional probability equation:

$$P_t(\pi) = P(A)P(B|A) \quad \ldots (14)$$

where, Pt (π) is the probability of tournament selection of chromosome π, P(A) is probability that the chromosome is selected in the first step of the tournament selection, and P(B | A) is probability that the chromosome is selected in the second step of tournament selection, assuming that it was previously selected in the first step.

Having in mind that the values of parameters n and p_t are given, it can be concluded that the probability of tournament selection of chromosome is linearly dependent on its rank, similar to the probability of selecting chromosomes by their rank. However, tournament selection is more efficient than the selection by the rank.[27] The time complexity of the described algorithm for selecting the rank is O(n log n), while the time complexity of tournament selection is O(n), because tournament selection does not require organizing the generation of chromosomes by their adaptability level. In addition to that, tournament selection allows parallelization.

Crossover

Two chromosomes π_r1 and π_r2 are selected from the current generation in one of the described ways. These chromosomes will be named parent chromosomes. In the basic version of the GA, parent chromosomes are cross-linked with probability p_c, where p_c represents the predefined parameter of the algorithm. Cross-linking of two parents generates two new chromosomes π_d1 and π_d2, which will be named offspring. If parents are crossed-linked, their offspring is transferred to a new generation; otherwise, parents are transferred to a new generation.

The crossover of chromosomes π_1 and π_2, defined by the Eqs (1) & (2) is illustrated by Fig. 6. The symbols of five randomly selected positions in parents transmitted to the descendants are framed. Arrows denote the transfer of symbols from parents to descendants.

Symbols on five randomly selected positions in parents transmitted to offspring are framed. Arrows mark the transfer of symbols from parents to offspring.

In example 2, the fitness of parents π_1 i π_2 to the given cipher text "vjururhapwhzcip" were calculated:

$f(\pi_1) = -70{,}74802657379654\ f(\pi_2) = -63{,}8493424363967$

The offspring fitness to the same cipher text are:

$$f(\pi_{d1}) = P\big(\pi_{d1}(jururhapwhzcip)\big) = P(mjtqtqprnwphuxn) = -76.53437220177032 \quad \dots (15)$$

$$f(\pi_{d1}) = P\big(\pi_{d2}(jururhapwhzcip)\big) = P(okdedexicmxbzlc) = -59.49717839615627 \quad \dots (16)$$

It can be noted that the fitness of chromosome π_d2 is higher when compared to parents, which implies the "better-quality" solution. This illustrates the basic idea of GAs that the chromosomes crossover results in chromosome fitness growing throughout generations. However, offspring can also be less fitted than parents, which is the case with chromosome π_d1, so there is a risk that quality solutions from a certain generation are lost during the evolution process. To avoid this, an elitist strategy is being applied, which foresees that the share of the most fitted chromosomes in the current generation is directly transferred to a new generation.

Experimental Results

The implementation of the GA in previously mentioned scenario is extremely sensitive to local maximums, so the appropriate choice of parameter values is important for the algorithm effectiveness. In general, there are non-linear dependencies between the GA parameters, and the parameter optimization cannot be performed independently of other parameters. There is currently no agreement on the optimal choice of parameter values[28], and their practical determination is usually based on the experience from previous successful application and experimentation. GA is executed until it meets the predetermined condition for terminating its execution. Defining termination conditions is a non-trivial task, and it depends on the problem that is being modelled. In the particular case, the fact that the maximum fitness of chromosomes in the generation "converges" can be used for the definition of termination conditions.

Fig. 6 — Crossover of chromosomes π_1 and π_2 and the generation of the first π_d1 and second π_d2 offspring

In Table 1 we provide the configuration parameters that we have used in our experimental work based on dataset that uses RSA, RC4, DES and AES encryption algorithms by Weka tool. Presented configuration of GA minimizes the classification error from kNN, keeping in mind also that best fitness and mean fitness should be close in value, in order that GA converges towards completion. High number of generations was used in order to increase the possibility of decrypting of data.

Throughout this paper, we have considered selected aspects of automatic cryptanalysis of simple substitution ciphers using statistical language models. The reason is practical - representative data on the frequency of letter n-grams in the English language are publicly available. However, it is important to note that the described approaches do not depend on the choice of language. Assuming that the plaintext language is known, and that representative data on the frequency of n-grams in the given language are available, the described approaches can be applied, regardless of whether the attacker knows the given

Table 1 — Parameters used in GA

GA parameter	Value
Population size	100
Genome length	100
Population type	Bitstrings
Fitness function	kNN — Based classification error
Number of generations	300
Crossover	Arithmetic crossover
Crossover probability	0.8
Mutation	Uniform mutation
Mutation probability	0.1
Selection scheme	Tournament of size 2
Elite count	2

language. There is no fitness function for combined algorithm to crack simple ciphers. Different cipher breaking algorithms use different fitness functions (i.e. for transposition cipher uses the bigram-trigram equation, for substitution cipher, it uses monogram equation, etc.). Keeping this in mind, we have proposed this fitness function for a combined algorithm consisting of different simple cipher breaking techniques. We have shown the relationship between key length and fitness value after 300 generations.

Obtained results were then imported in Weka tool again, and deep learning techniques were used to validate the applicability of GAs for cryptanalysis. The aim of this validation was to check whether GAs were capable in the process of deciphering of encrypted text. Elements from dataset were divided into 5 groups for the purpose of multiple iterations and verification of results over training and test data. Results from Table 2 tell us that approximately 95% of cipher text can be accurately decrypted by using GAs based cryptanalysis for lightweight block ciphers such as RC4 and DES, while AES and RSA shows better resistance to cryptanalysis with only 65% of decrypted text. Literature resources[21,22,24] indicate that widely used algorithm in machine learning is the artificial neural networks, which try to simulate the behavior of the human brain for decision-making. A very simple type of network is called Multilayer Perceptron (MLP), which are networks with an input layer, one or more intermediate layers, and an output layer. Deep Learning networks are a new type of neural network that discovers important object features. These networks determine features without supervision, and are adept at learning high level abstractions about their data sets. All previously stated is the reason why we exploited them in this paper.

Table 2 — Applicability of GA based approach for deciphering of data

No.	Cipher algorithm + dataset part	Accuracy (%)
1	RC4 — dataset part 1	96.02
2	RC4 — dataset part 2	94.35
3	RC4 — dataset part 3	97.70
4	RC4 — dataset part 4	95.55
5	RC4 — dataset part 5	95.09
6	DES — dataset part 1	97.73
7	DES — dataset part 2	95.01
8	DES — dataset part 3	96.15
9	DES — dataset part 4	94.00
10	DES — dataset part 5	95.89
11	AES — dataset part 1	65.86
12	AES — dataset part 2	62.32
13	AES — dataset part 3	67.58
14	AES — dataset part 4	63.47
15	AES — dataset part 5	66.76
16	RSA — dataset part 1	68.15
17	RSA — dataset part 2	65.52
18	RSA — dataset part 3	63.28
19	RSA — dataset part 4	64.71
20	RSA — dataset part 5	65.36

Discussions

As we already know, GAs represent a category of natural computing algorithms, which emulate phenomena and behaviors observed in nature. These algorithms are applied to tackle optimization problems where the goal is to discover optimal values within a defined search space, either maximizing or minimizing depending on the problem under examination.[22,26,28] This paper introduces a fitness function that holds relevance to the application of cipher breaking algorithms. However, the prospect of enhancing problem-solving methods for cryptanalysis lies in harnessing the potential of a random function generator, thereby reducing time complexity. The random function generator can evolve beyond its role as a key generator for each iteration and be further investigated to specialize its function. In this context, the significance of efficient crossover and mutation operations cannot be overstated, as they substantially influence algorithm performance.

To elevate the utility of link grammar and the fitness function, an efficient technique for generating candidate sentences from decrypted text stands as a cornerstone. Moreover, the word graph should be adapted to accommodate a restricted set of deliberately appended non-sense words, which the sender might introduce. This modification ensures that the word graph remains a robust tool in cryptanalysis.

Conclusions

In this paper, some useful analogies between the cryptanalytic systems and the machine learning systems were given. Great success of training systems with deep architecture, as well as the existing self-training phase of pre-training provides greater possibilities of applying these techniques in cryptanalysis. We have shown and experimentally validated the potential of application of GAs for cryptanalysis. The main advantage of applicability of GAs lies in the fine-tuning of the fitness function of GA which can help in achieving of better results. The future research in the field of

cryptanalysis as well as further development in the area of deep leaning will demonstrate whether the classic criteria for the cipher system quality might be joined by some addition regarding to the area of machine learning, such as the depth of equivalent cipher transformation architecture or the training sets size in the pre-training and training phases.

References

1 Rosenheim S, "The king of `secret readers'": Edgar Poe, *ELH*, **56(2)**, (1989) 375–400, https://doi.org/10.2307/2873064.

2 Jaćimovski S & Šetrajčić J, Physical fundamentals of quantum cryptography, *Proc Archibald Reiss Days*, (2016) 276–292, Belgrade, Serbia.

3 Sony J & Goodman R, A mind at play: How Claude Shannon invented the information age, *Simon & Schuster*, (2017) New York, USA.

4 Schieler C & Cuff P, Rate-distortion theory for secrecy systems, *IEEE Trans Inf*, **60(12)** (2014) 7584–7605, https://doi.org/10.1109/TIT.2014.2365175.

5 Prajapat S & Thakur S R, Various approaches towards cryptanalysis, *Int J Comput Appl*, **127(14)** (2015) 15–24, https://doi.org/10.5120/ijca2015906518.

6 Ali H F, Al-Safi G S M & Yousif A A, Analyzing cryptosystems by using artificial intelligence, *J Sci*, **1(1)** (2018) 100–108, https://doi.org/10.22401/ANJS.00.1.14.

7 Lafitte F, CryptoSAT: a tool for SAT-based cryptanalyais, *IET Inf Secur*, **12(6)** (2018) 463–474, https://doi.org/10.1049/iet-ifs.2017.0176.

8 Chizhov I & Tashevtseva N, Logical cryptanalysis as part of strength research into particular code based signature scheme, *Int J Open Inf Technol*, **8(10)** (2020) 27–38.

9 Semenov A, Otpuschennikov I, Gribanova I, Zaikin O & Kochemazov S, Translation of algorithmic descriptions of discrete functions to sat with applications to cryptanalysis problems, *Log Methods Comput Sci*, **16(1)** (2018) 29:1–29:42, https://doi.org/10.23638/LMCS-16(1:29)2020.

10 Nejati S & Ganesh V, CDCL (Crypto) SAT solvers for cryptanalysis, *Proc of the 19th CASCON*, (2020), 1–6, New York, USA, https://doi.org/10.48550/arXiv.2005.13415.

11 Alani M M, Applications of machine learning in cryptography: A survey, *Proc of ICCSP '*19 (2019) 23–27, https://doi.org/10.1145/3309074.3309092.

12 Haber E & Ruthotto L, Stable architectures for deep neural networks, *Inverse Probl*, **34(1)** (2017) 1–23, https://doi.org/10.1088/1361-6420/aa9a90.

13 Basu S, Karki M, Ganguly S, DiBiano R, Mukhopadhyay S & Nemani R, Learning sparse feature representations using probabilistic quadtrees and deep belief nets, *Neural Process*, **45(3)** (2015) 855–867, https://doi.org/10.1007/s11063-016-9556-4.

14 Becker M, Lippel J, Stuhlsatz A & Zielke T, Robust dimensionality reduction for data visualization with deep neural networks, *Graph Models*, **108** (2020) 1–15, https://doi.org/10.1016/j.gmod.2020.101060.

15 Lee R T, Sen Teh J, Suet Yan L J, Jamil N & Yeoh Z W, A machine learning approach to predicting block cipher security, *Proc 7th Int Cryptol Inform Secur Conf* (Serdang, Malaysia) 2020 122–132.

16 Al-Janabi T S, Al-Khateeb B & Abd J A, Intelligent techniques in cryptanalysis: Review and future directions, *UHD J Sci Tecnol*, **1(1)** (2017) 1–10, https://doi.org/10.21928/uhdjst.v1n1y2017.pp1-10.

17 Andonov S, Dobreva J, Lumburovska L, Pavlov S, Dimitrova V & Popovska-Mitrovikj A, Application of machine learning in des cryptanalysis, *Proc 12th ICT Innov Conf* (Skopje, North Macedonia) 2020, 1–11.

18 So J, Deep learning-based cryptanalysis of lightweight block ciphers, *Secur Commun Netw*, **2020** (2020) 1–11.

19 Taran O, Rezaeifar S & Voloshynovskiy S, Bridging machine learning and cryptography in defence against adversarial attacks, *Proc 10thEuropean Conf Comput Vision*, 2018, 267–269, https://doi.org/10.48550/arXiv.1809.01715.

20 El-Zoghabi A A, Yassin H A & Hussein H H, Survey report on cryptography based on neural networks, *Int J Emerg*, **3(12)** (2013) 465–472.

21 Kim H, Lim S, Kang Y, Kim W, Kim D, Yoon S & Seo H, Deep-learning-based cryptanalysis of lightweight block ciphers revisited, *Entropy*, **25(7)** (2023) 986, https://doi.org/10.3390/e25070986.

22 Blackledge J, Bezobrazov S & Tobin P, Cryptography using artificial intelligence, *Proc IJCNN*, (Killarney, Ireland) 2015, 1–6, https://doi.org/10.1109/IJCNN.2015.7280536.

23 Liu H, Kadir A & Xu C, Cryptanalysis and constructing s-box based on chaotic map and backtracking, *Appl Math Comput*, **376** (2020) 125–153, https://doi.org/10.1016/j.amc.2020.125153.

24 Pan J, Encryption scheme classification: A deep learning approach, *Int J Electron Secur Digit Forensics*, **9(4)** (2017) 381–395, https://doi.org/10.1504/IJESDF.2017.087397.

25 Ahmad U, Song H, Bilal A, Alazab M & Jolfaei A, Secure passive keyless entry and start system using machine learning, *Proc Int Conf Secur Privacy Anonymity Comput Commun Storage*, (2018) 304–313, Melbourne, Australia, https://doi.org/10.1007/978-3-030-05345-1_26.

26 Saif A & Rabidi M R, Machine learning based attack on certain encryption schemes, *Proc of the 2nd Int Conf Comput Appl Info Secur* (Riyadh, Saudi Arabia) 2019, 1–6, https://doi.org/10.1109/CAIS.2019.8769527.

27 Garg P, Varshney S & Bhardwaj M, Cryptanalysis of simplified data encryption standard using genetic algorithm, *J Commun Netw*, **4(3)** (2015) 32–36, https://doi.org/10.11648/j.ajnc.20150403.12.

28 Forhad A S, Hossain S & Rahman O M, An improved fitness function for automated cryptanalysis using genetic algorithm, *Indones J Electr Eng Comput Sci*, **13(2)** (2019) 643–648, https://doi.org/10.11591/ijeecs.v13.i2.pp643-648.

29 Bulatović L, Mijanović A, Asanović B, Trajković N & Božović N, Automated cryptanalysis of substitution cipher using Hill climbing with real designed heuristic function, *Math Montisnigri*, **44** (2019) 135–143, http://doi.org/10.20948/mathmon-2019-44-11.