Iranian Journal of Fuzzy Systems

Volume 21, Number 1, (2024), pp. 33-49

Original paper

University of Sistan and Baluchestan

A fuzzy optimal lightweight convolutional neural network for deduplication detection in cloud server

J. K. Periasamy [1], L. Selvam [2], M. Anuradha [3] and R. Kennady [4]

[1] *Department of Computer Science and Engineering,Sri Sairam, Engineering College,Chennai, India*
[2] *Department of Information Technology, K S R Institute for Engineering and Technology, Tiruchengode, Chennai-637215, Tamil Nadu, India*
[3] *Department of Computer Science and Engineering, S.A. Engineering College, Thiruverkadu, Chennai-600077, Tamilnadu, India*
[4] *Department of Artificial Intelligence and Data Science, Rajalakshmi Institute of Technology Kuthambakkam, Chennai, 600124, Tamil Nadu, India*

periasamyjk.mail@gmail.com, selvaml@inbox.lv, anuradham@inbox.lv, rkennady@inbox.lv

Abstract

Nowadays the cloud computing environment is widely utilized for transmitting and receiving data securely. Inorder to secure the data the encryption method is used but still due to some limitations the security process is diminished. Therefore, this paper proposes a new algorithm to provide better security while transmitting data through the network. At first, the sensitivity of data is determined using a lightweight convolutional neural network (LWCNN) model which is used to categorize the unclassified data into two categories normal sensitive data and highly sensitive data. After determining the level of data sensitivity, the encryption process is performed further. The efficient hash function-based duplication detection approach is employed to maintain confidential information before outsourcing it to a cloud server. Subsequently, the ideal keys are generated for each data based on its sensitivity level using the proposed fuzzy tuna swarm (FTS) algorithm. Finally, the data is encrypted by converting plain text into ciphertext which is only visible to authorized users. The experimental results show that the LWCNN model utilized for data sensitivity classification achieved 94% accuracy and the FTS algorithm proposed for optimal key generation took much less communication time of about 1800 μs than other compared techniques.

Keywords: Security, encryption, key generation, lightweight convolutional neural network, fuzzy logic system, tuna swarm optimization algorithm.

1 Introduction

Due to the expansion of data in an uncontrolled manner, it is necessary to increase the memory of digital data. The data de-duplication removes unnecessary data and it creates the device storage with more space to share the networks. It reduces the size of the data to improve the storage in the device [1]. The shared file may be classified into various sub-data and it helps the user to evaluate and select the particular data. The unnecessary classified data are determined and obtained in cloud computing. The stored unnecessary data are used when there occurs a need for matching between the classified data. The process of data storing diminishes the consumption of time [14]. In public organizations and transportation departments, more digital images are circulated periodically. Many images are transformed for the user by cloud computing and it provides more space to store the data. Image privacy is maintained by encrypting the images before transferring them [11]. The two types of compression methods are data de-duplication and data reduction. In the data reduction model, the excess data are noticed by using the dictionary model. But in data de-duplication, the part of the classified data is stored separately and if needed it could be selected from the saved location. The data

Corresponding Author: J. K. Periasamy
Received: May 2023; Revised: October 2023; Accepted: November 2023.
https://doi.org/10.22111/IJFS.2023.45797.8070

de-duplication method combines two data namely duplication and resemblance which helps to reduce the size [18]. After the development of digital technology, the security of various data has been affected. To overcome the issue the security devices create various statistical and computational methods to protect from illegal activities. The computer privacy system namely steganography and cryptography is generated. The encryption and decryption obtained in cryptography are used to secure old data communication in which no other person has any idea about the information [3]. The IoV application obtained in 5G communication is applicable in real-time traffic congestion and navigation. The insufficiency issue is solved by the edge computing method. The ABE technique in the encryption method can communicate with multiple vehicles but in decryption, the combining of various data is expensive [8]. The information shared among two users is secured by a method called cryptography. The efficiency of the Data Encryption Standard (DES) is minimized and it is applied only in data encryption. The encryption method is divided into two types namely pixel diffusion and pixel permutation. The pixel position is varied in pixel permutation and the value of intensity is varied in pixel diffusion [13]. The introduction of the Internet of Medical Things (IoMT) method is applied in medical imaging which provides a facility for doctors to provide treatment. The encryption and decryption method is applied in the medical field to secure the data of the patients and IoMT. To overcome the hardness obtained in IoMT, the CNN technique is generated in image processing methods. The image-to-image transformation is performed by Deep-learning-based Encryption and Decryption Network (DeepEDN) [7]. The AES (Advanced Encryption Standard), DES (Data Encryption Standard), and IDEA are some of the encryption methods used for security issues due to image decryption. The chaos-based system present in the encryption process could not generate the same data repeatedly [2]. The arithmetic and logical calculations are done in the encryption method by a Fully Homomorphic Encryption (FHE) scheme. The FHE techniques are combined with various algorithms to increase the performance of various core data with the help of FPGA [12]. Although there exist many encryption algorithms to offer better security, they still hold certain limitations. Therefore, this paper proposes a new algorithm to provide better security while transmitting data through the network. The key contributions of this paper are described as follows:

- A novel FTS algorithm is proposed for optimal key generation based on the data sensitivity level.
- The sensitivity level of unclassified data is classified into two categories as normal sensitive data and highly sensitive data using the LWCNN model.
- Encrypting confidential information before transmitting it to the cloud server using an efficient hash function-based deduplication detection algorithm.
- Investigating the performance of the proposed algorithm is in terms of different measures namely classification accuracy, communication time, success rate, memory utilization, average delay, communication overhead, computation time, encryption time, and throughput.

The rest of the paper is as follows. The existing works of various authors are described in Section 2. The proposed method for data encryption is explained in Section 3. Section 4 discussed the results and discussion of the model and Section 5 concludes the article.

2 Related works

Periasamy et al. [15] developed an Efficient Hash Function-based Duplication Detection (EHFDD) algorithm to perform the detection of duplicate data. Addressing and eliminating the duplicate data was the main challenging task in this paper. The EHFDD approach was introduced in this paper to provide security for data and report the problem related to the reduction detection of authorized data. As an experimental result, the introduced approach attained optimized better performance for removing the duplicate data which is used for a cloud server to minimize the memory storage capacity of 8% and maximize the bandwidth. EHFDD approach enhances the accomplishment rate of .13%. Meanwhile, the introduced approach takes more space to store data. Ahmed et al. [1] established the Detection of resistive patterns as fragment separators using the Lightweight triple-level hashing (LT-LH) function. One of the main objectives of the paper is to reduce data duplication. To shortcoming this problem, LT-LH was developed in this paper. To validate the activity of the LT-LH, numerous tests to analyze the characteristics of the data. Therefore, the performance of the developed approach leads to an increase in the percentage of eliminating duplicate data and minimizes the data the system requires. The LT-LH approach achieved superior performance for saving storage five times faster than the SHA and MDA approaches. However, these approaches ignore the null values.

Li et al. [10] elaborated on a reduction of hashing for remote senor image retrieval using the Quantized deep learning (QDL) framework. The Convolution neural network was employed to develop a hashing algorithm but it requires more storage space and is also more expensive. To overcome this problem, the QDL framework was introduced in this paper. It needs only a small amount of storage space and fewer computing resources. As a result, the developed method enhanced the efficiency. On the other hand, in some other models like BERT, it is possible to lose accuracy. Periasamy et al. [14] reviewed the concept of an enhanced secure content duplication identification and prevention (ESCDIP) algorithm for cloud computing. Classifying the duplicate data on a cloud server and reducing the storage capacity at a low cost were the main purposes of this paper. ESCDIP algorithm was attained to classify the individual data fragmentation for storing cloud servers. Therefore, the ESCDIP technique minimizes 2.3% of data uploaded time in seconds compared to other previous methods. However, the developed ESCDIP approach does not guarantee reliability and consistency.

Ding et al. [6] explained medical image encryption and decryption using deep learning-based key generative adversarial network (DL-GAN). The generator of stream cipher is utilized to create the private key to provide security for the patient's medical dataset. The main purpose of the DL-GAN method was to transfer the original images to the private key. Two datasets were utilized to validate the performance of the DL-GAN as an ultrasonic brachial plexus data set, an X-ray dataset. Therefore these developed approaches achieved generating high security in the private key. On the other hand, these approaches are possible for data overfitting. Zeebaree et al. [20] reviewed the encryption and decryption of the data encryption standard (DES) algorithm to implement the field programmable gate array (FPGA). Secure data on the internet was the most important challenging task. FPGA is used to provide higher security of data. As a result, the developed approach achieves a higher throughput rate compared to other hardware executions. On the other hand, this method fails due to the slow detection of duplicate data.

Imran et al. [9] illustrated decryption and encryption for speech signals using the EI-Gamal algorithm. Securing data information on the wireless communication network (WCN) and the Internet is the main objective of this paper. Cryptography is used to protect data and also convert the data from readable to unreadable. Therefore, the output of the introduced algorithm achieved good-quality speech signals compared to the initial signal. Meanwhile, these approaches do not accept video signals and color images. Ding et al. [7] explained Image encryption and encryption for medical information using Deep Learning (DL). One of the main aims of this paper is to secure the medical data information of the patients. Cycle generative adversarial network (cycle-GAN) was introduced in this paper to provide security among medical data information for transferring their original area to the targeting area. Chest X-ray was utilized to evaluate the process of image encryption and decryption. Hence, the developed approach optimized a superior higher security level. Meanwhile, training on complex data is difficult.

3 Proposed methodology

One of the biggest challenges faced by distributed computing systems is data security. In the digital world, data security has received major concern in the cloud computing environment for the secure transmission of sensitive and confidential information through the internet. Encryption is the best and most effective way to protect data from third parties. Although there exist many encryption algorithms to offer better security, integrity, and authenticity, they still hold certain limitations. Encryption algorithms are becoming more essential nowadays for secure data reception and transmission from the cloud environment. The data encryption process encodes the data and ensures better security even if the information is leaked.

The FTS method is proposed that generated an ideal key to determine the sensitivity level of data transmission. The determination of the LWCNN model designed the resolution of images and the distinct features are targeted to execute the test and trial of suitable networks. It also distinguished the unclassified data into normal as well as highly sensitive data. The sensitive data are encrypted before transmitting to the cloud server using an efficient hash function-based deduplication detection algorithm. This process also removes redundant information and unused memory space. Based on their sensitivity level, optimal and unique keys are generated for each data using the proposed FTS algorithm.

Taking inputs namely plaintext and key as input, the encryption process generates ciphertext. These procedures guarantee well-secured data transmission in a cloud environment against unauthorized access. The structure of the proposed secure transmission model is presented in Figure 1.

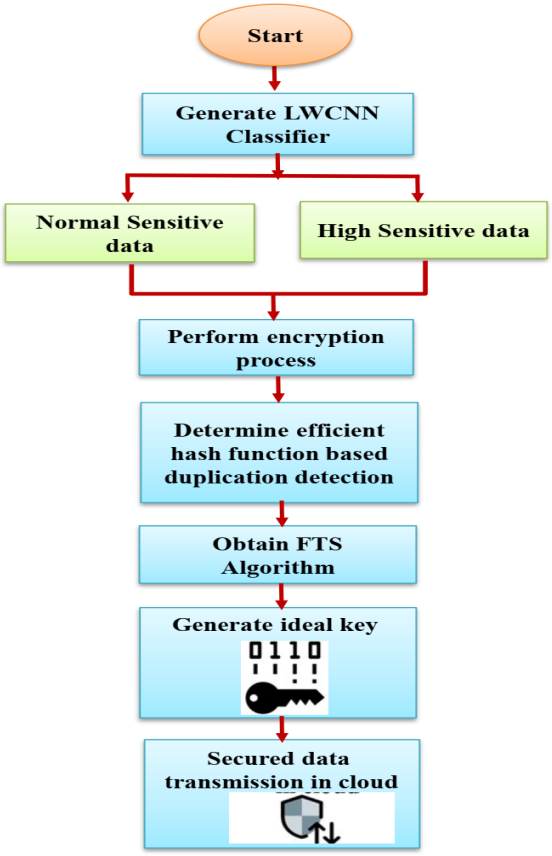

Figure 1: Structure of the proposed secure transmission model

3.1 Data classification

Assume 1GB of data and send it to the cloud to explain the importance of data classification. Among this 1GB data, 200 MB is considered highly sensitive data and the remaining data is said to be normal sensitive data [4]. A powerful algorithm is used for the encryption of 1GB of data that increases the time. In the cloud and the client, the classification of data has been employed. Moreover, L-CNN is used for categorizing the data into two classes: Highly sensitive data and Normal sensitive data.

I. Highly sensitive data

Highly sensitive data is risky data when the data has been revealed illegally or the data has been destroyed. It affects the users and makes them tense about losing their data. Here data is referred to as personal data, military information, business records, financial material, and government information.

II. Normal Sensitive data

Normal sensitive data is a risk-free organization even if data is revealed illegally or any destruction happens. Here the data is referred to as videos, introductory information, photos, marketing material, or organization's press announcements.

3.1.1 Lightweight CNN design

Convolution neural networks (CNN) are utilized for the classification and detection of the object. The VGGNet, AlexNet, DenseNet, ResNets, and GoogLeNet are the CNN networks and these networks are utilized for object detection, picture classification, and character recognition [5]. But these networks are not suitable for the detection of a tiny moving vehicle due to the following reasons;

1. The CNN is not modeled for high-resolution as well as natural pictures. The targets have distinct features and are larger in the images. The deep networks are utilized for extracting the features with higher semantic information. On the other hand, the targets are really small in the satellite videos.
2. A deep CNN network requires a higher computational cost and lower efficiency. Faster computation is required in video processing.

The lightweight CNN (LCNN) networks are designed for the detection of tiny vehicles with satellite videos. The networks are inspired by the utilization of the LeNet5 network.

LCNN architecture

Generally, there is no rules are followed to design CNN but it depends on user experience as well as the complexity of the model. The greatest way to find suitable networks is by executing the test and trial. The lightweight CNN contains 1 convolutional layer and it is followed through a max-pooling layer. In other lightweight approaches, the overfitting is maximized while performing classification. This diminished the classification performance and can detect the duplicated data accurately. The LCNN model distinguishes the unclassified data into normal and highly sensitive data as well as determines the sensitivity level of each transmitted data in a cloud server. It utilized different networks to classify the individual characteristics. The execution of test and trial is performed in each data for finding the specific network to design CNN.

Convolutional layer

One of the core building blocks of CNN is the convolution layer utilized for extracting the input image features. The CNN networks adopted 8 filter channels, which performed better for the dataset. However, 7,5, and 3 are considered the best size of the image filter. In the testing process, 5 is indicated as the best size. In LeNet5, the sigmoid activation function is utilized after the convolutional layer.

Max pooling

The average pooling is utilized in LeNet5 and after the convolution layer; the max-pooling layer is utilized. After the pooling, the 8 feature map is resized as 8@16x16.

Fully-connected layer

The feature map is flattened to the vector with a size of 2048 as well as 100 neurons are employed in a fully connected layer.

Output layer

It divides the classes into dual layers but determines two neurons with moving vehicles.

Loss function

The loss function is utilized for measuring the performance of neural network design. The detection of the moving vehicle is considered a classification issue. The output is softened by the SoftMax activation function is adapted for the normalization of output probability. The total amount of loss function is also known as binary cross entropy and it is expressed as;

$$T(\theta) = -\frac{1}{\eta}\sum_{j=1}^{\eta}\left[z_j \; log \;\; (\rho_j) + (1 - z_j)\, log\,(1 - \rho_j)\right]. \tag{1}$$

From the above equation, the number of samples is indicated by η, index observation or samples are represented by j, and the true labels for jth sample are denoted by z_j.

3.2 Efficient Hash function based duplication detection (EHFDD) algorithm

The utilization of the EHFDD algorithm maps the random dimension or the permanent dimension of the dataset. The return values of the hash variables are determined by hash sums, hash codes, and hash values and it is suitable for the isolation of data in de-duplication. In this paper, the proposed method blocked the older encrypted message and it analyzed the space-efficient data stored in the outsourced memory. The problems obtained in block-level de-duplication and key management are solved and it is applied in the system software which recaptures the data easily. The Hash function is used to determine the duplicate records that are obtained in the large file repeatedly.

It is also applied in the cryptography method which validates whether the data is obtained in the hash function or not and it also signifies the undetected input data that are stored in the hash value. Each factor is designed for specific requirements and it contains various designs and optimization frameworks. It plays a challenging role in the bloom filter to determine the component whether is obtained in the member set by the bloom filter [1]. The hash function is simulated in tag generation and it estimates the stored value W with the help of a cryptographic factor $R_W = F(W)$. The encryption data R_W is accelerated by the secret key R_q to ignore the deterministic key generation factor. The expression for the encryption key is formulated as

$$R_{W,q} = F_0\left(F(W)R_q\right) \oplus F_2(W). \tag{2}$$

The cryptographic hash functions are denoted by F_0 , F_1, and F_2. The developed method improves the Storage Cloud Service Supplier (S-CSS) and private cloud server that will not decrypt the cipher text.

System setup

The assimilation of the protocol is written as

$$\Gamma = (P, V). \tag{3}$$

The POW is integrated with the hash function F, F_0, F_1 and F_2. Each identity of the client is stored in the cloud server separately.

Proof of ownership (PoW)

A specification for ownership proof notices in deduplication systems that an owner can credibly demonstrate that the client has a data file on a cloud server without uploading it to the cloud. Several PoW constructions are established in owner-side deduplication based on the Merkle-hash tree which has a limited leakage system. Another method of proof of ownership facilitates the arbitrary choice of a file's bit state as the source of the file.

Uploading a file

If a file F_e is uploaded and shared with multiple users belonging to the collection $P_e = P_{e_n}$, then the data owner carries out the documentation and transmits it to the private cloud server $M(F_e)$. Two sets of file tags are represented as follows

$$\varphi_{F_e,P_e} = M_0(M(F_e)S_{P_e}). \tag{4}$$

$$\varphi'_{F_e,P_e} = M_1(M(F_e)S_{P_e}). \tag{5}$$

Here, every P_e, $\Re(P, P_e)$=1, will be returned to the user along with the identity cards.

3.3 Key generation

In multimedia data communication applications, data security becomes an important aspect due to increased unauthorized access to confidential data. In order to protect private information from such invaders, an effective cryptographic mechanism is required. The generation of optimal keys restricts the invaders from accessing valuable information by converting it to an unreadable format. Therefore, in this paper, we designed a novel FTS algorithm for the generation of optimal keys.

3.3.1 Fuzzy logic system

The fuzzy set is referred to as the membership function whose values are either one or zero [17]. If the membership function falls to zero then the object does not belong to the set and if it is one then it is determined with the subset. The fuzzy logic control is separated into three phases and they are fuzzification, rule evaluation as well as defuzzification.

Fuzzification

The fuzzy logic rules are determined as;

Rule 1: If M is γ_1 and N is ν_1, then $\Re$ is τ_1.

Rule 2: If M and N is γ_2 and ν_2, therefore $\Re$ is τ_2.

Where the conditional variables are denoted by M and N, $\Re$ is the response variable, fuzzy metrics characterized through membership functions and it is represented by γ_j, ν_j, τ_j. Let E_1 and E_2 are considered as intervals, hence conditional variables are evaluated as $\varphi_{\gamma 1}(n)$ and $\varphi_{\nu 1}(m)$ in rule 1 as well as $\varphi_{\gamma 2}(n)$, $\varphi_{\tau 2}(m)$ in rule 2.

Rule evaluation

The control rule satisfies the condition $M = n$ and $N = m$ it is expressed as;

Rule 1: $\varphi_1 = \varphi_{\gamma 1}(n) \wedge \varphi_{\nu 1}(m)$;

Rule 2: $\varphi_2 = \varphi_{\gamma 2}(n) \wedge \varphi_{\nu 2}(m)$.

The Mamdani-type rule evaluation is denoted by $\wedge$, and truth degrees are indicated by φ_1 and φ_2. The membership $\varphi_{\tau j}(\Re)$ of output is defined as the summation of memberships and it is expressed as;

$$\varphi_{\tau}(\Re) = \varphi_{d1}(\Re)^{*} \varphi_{d2}(\Re). \tag{6}$$

Where $*$ indicates the max function of the Mamdani-type rule.

Defuzzification

The outcome of the result is transformed into real values and determined in fuzzy logic inputs. The Area Of Gravity (COA) approach is adopted and is expressed as;

$$Z_{COA} = \frac{g_a \varphi_A(Z) Z dz}{g_a \varphi_A(Z) Z d}. \tag{7}$$

From the above equation, the algebraic integral based on the fuzzy subset Z is indicated by g_a.

3.3.2 Tuna swarm optimization (TSO) algorithm

a. Inspiration:

The tuna is a remarkable and admiring carnivorous fish that lives in seawater. The scientific name of tuna is Thunnini which means marine carnivorous fish [19]. There are varieties of tuna species that have been present and their size differs greatly. This predator catches the prey from surface water and midwater. This predator has its own way of swimming, in which the tail swings faster while the body keeps still. Though tunas are very fast swimming predators, still nimble small fishes are faster. So, the tuna goes as a gang to attack the prey with brilliant foraging tricks. There are two methods used by this predator. They are

Spiral foraging

These predators direct the prey into the shallow water by forming a spiral-shaped formation. Thus, the predators catch the prey without risk.

Parabolic foraging

Every tuna swims one after the other and forms a parabolic shape to surround the prey. Thus, the Tuna catches the prey successfully with the above two methods. Based on these two methods, a new swarm-based metaheuristic optimization algorithm is utilized.

b. Analytical model:

The analytical model of the TSO algorithm is explained below.

Initialization

In the search space, the initial population with random uniform has been produced by starting the TSO process and is mathematically expressed below.

$$Y_j^i = random\ .\ (B_u - B_l) + B_l, \qquad j = 1, 2, \cdots, m, \tag{8}$$

Where m denotes the tuna population's number, $random$ is the random vector, B_l and B_u represents the lower and upper boundary, Y_j^i is the initial individual of j^{th}.

Spiral Foraging:

Each fish has its own way of swimming strategies and forms a group to protect itself from predators. Though other fishes know the strategies, tuna adjust according to the prey's direction and form a dense spiral formation for catching the prey. Tuna gathers the information from the foregoing fish and spreads it to the whole group of fish. The statistical expression of the spiral foraging strategy is given in the below equation.

$$Y_j^{s+1} = \begin{cases} \beta_1.\ \left(Y_e^s + \alpha.\ |Y_e^s - Y_j^s|\right) + \beta_2.Y_j^s, & j = 1, \\ \beta_1.\ \left(Y_e^s + \alpha\ .|Y_e^s - Y_j^s|\right) + \beta_2.Y_{j-1}^s, & j = 2, 3, \cdots, m, \end{cases} \tag{9}$$

$$\beta_1 = c + (1 - c)\ .\frac{s}{s_{\max\ imum}}, \tag{10}$$

$$\beta_2 = (1 - c) - (1 - c)\ .\frac{s}{s_{\max imum}}, \tag{11}$$

$$\alpha = e^{bl}.\ \cos\ (2\pi b), \tag{12}$$

$$l = e^{3\cos(((s_{\max imum} + 1/s) - 1)\pi)}. \tag{13}$$

From the above equation, a is a random number, β_1 and β_2 represents weight coefficients, s signifies current iteration, Y_j^{s+1} is the j_{th} individual of the $s + 1$ iteration, c denotes constant, $s_{\max imum}$ represents maximum iteration. Every individual searches for more space with the ability to explore and make TSO.

$$Y_j^{s+1} = \begin{cases} \beta_1.\left(Y_{random}^s + \alpha.|Y_{random}^s - Y_j^s|\right) + \beta_2.Y_j^s, & j = 1, \\ \beta_1.\left(Y_{random}^s + \alpha.|Y_{random}^s - Y_j^s|\right) + \beta_2.Y_{j-1}^s, & j = 2, 3, \cdots, m, \end{cases} \tag{14}$$

Where Y_{random}^s represents a random reference point. In spiral foraging, the changes of optimal individuals from random individuals take place. It is done by TSO when the iteration increases. The analytical form of the final spiral foraging system is given below.

$$Y_j^{s+1} = \begin{cases} \beta_1.\left(Y_{random}^s + \alpha.|Y_{random}^s - Y_j^s|\right) + \beta_2.Y_j^s, & if\ rand\ om < \frac{s}{s_{maximum}} \\ \beta_1.\left(Y_{random}^s + \alpha.|Y_{random}^s - Y_j^s|\right) + \beta_2.Y_{j-1}^s, & j = 1, 2, 3, \cdots, m, \\ \beta_1.\ \left(Y_b^s + \alpha.\ |Y_b^s - Y_j^s|\right) + \beta_2.Y_j^s, & j = 1,\ \ if\ random \geq \frac{s}{s_{maximum}} \\ \beta_1.\ \left(Y_b^s + \alpha\ .|Y_b^s - Y_j^s|\right) + \beta_2.Y_{j-1}^s, & j = 2, 3, \cdots, m, \end{cases} \tag{15}$$

Parabolic Foraging

In the feeding process, the tunas form another form i.e... Parabolic formation along with reference point (food).

$$Y_j^{s+1} = \begin{cases} Y_b^s + random.\left(Y_b^s - Y_j^s + f_t.q^2.\left(Y_b^s - Y_j^s\right)\right), & if\ random < o.5, \\ f_t.q^2.Y_j^s, & if\ random \geq 0.5, \end{cases} \tag{16}$$

$$q = \left(1 - \frac{s}{s_{\max imum}}\right)^{s/s_{\max imum}}, \tag{17}$$

From the above equation, f_t represents a random number. In the search space, one foraging strategy has been chosen for the regeneration of position. Finally, all the individuals are continually commutated and updated till the last condition is. Then the simultaneous fitness value and optimal individual are returned.

3.3.3 *Fuzzy tuna swarm (FTS) algorithm*

The FTS algorithm designed for the optimal key generation process creates optimal keys for the data and the workflow of the FTS algorithm is presented in Figure 2. The fuzzy logic system is used in the security process for selecting the membership function to determine an ideal key. It also determines whether the secured data is determined in the required subset or not.

The integration of fuzzy with the tuna swarm algorithm is determined to design the optimized key and update weight coefficients. The fitness value is computed based on weight coefficients. The ideal key of each data is selected and the sensitivity level by performing an encryption process to improve the security of data. For the effective selection of optimal keys, the fuzzy logic system is utilized in the operation of search schemes of the TSO algorithm. The key generation process commences by initializing the tuna population randomly in the search space. After initializing the population, the control variables c and x of TSO algorithm are assigned appropriate values.

The fitness value of each tuna is determined and the best initial random tuna is sorted. The weight coefficients are then updated which helps to control the movement of the tuna population toward the optimal solution. By exploiting and exploring the search dimension through spiral foraging and parabolic foraging schemes, 'n' solutions (ie. keys) are obtained which represent the best solutions with high fitness values. From the obtained solutions, the best solution (ie. optimal key) is determined by the application of the fuzzy logic system.

It not only selects optimal solutions but also avoids premature convergence problems that occur due to the random population initialization process. Thus, the proposed FTS algorithm creates/generates ideal keys for each confidential data. The ideal key generation process performed using the proposed FTS algorithm is presented in algorithmic steps on pseudocode 1.

3.4 Data encryption process

The main three components of encryption are key management, encryption engine, and data. The encryption algorithm is encrypted by securing the data [16]. The sender decides which kind of algorithms to be utilized as well as the variables utilized as the key. Therefore, the encrypted data is decrypted by utilizing the proper key distributed through the sender.

By the utilization of the secret key, the original data is encrypted through the user and created $\rho\nu_{\kappa-op}$, and $\rho\gamma_{\kappa}$ as well as the cipher data is transmitted to the server $\kappa = (\rho, r)\, en\,(m, \rho)$, and $\rho\nu_{\kappa} = (\kappa, j)$ for the random variable $\gamma \in Z_{\kappa}^{*}$, the cipher data is computed as $d = \tau.\gamma^{\kappa} MO\kappa^2$.

4 Experimental evaluation and discussions

In this section, the FTS algorithm is used for the key generation and LCNN is utilized for data classification. The performance evaluation measures namely throughput, memory utilization, encryption time, computational time, communication overhead, success rate, and average delay are determined for performing the validation process. The results for key generation and data classification are discussed below sections.

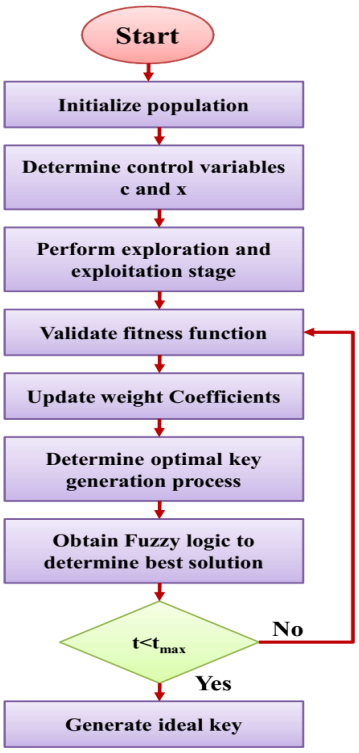

Figure 2: Flow diagram of proposed FTS algorithm

4.1 Experimental setup

The experiments are conducted on 4 GB RAM, Intel (R) Core (TM) i5-5200U CPU, and a processor speed of 2.20 GHz. The frontend system uses Visual Studio 2010 and ASP.NET along with C# and the backend uses Microsoft SQL Server 2008. The software is encrypted based on different sizes from 1 to 50 MB.

4.2 Hyperparameter configuration

The validation of hyperparameter tuning determined the optimal parameter values to estimate the performance of the proposed model. Table 1 depicts the hyperparameter values.

Table 1: Evaluation of hyperparameter values

Methods	Parameters	Ranges
LCNN	Learning rate	0.001
	Exponential decay rate for the first moment	0.9
	Exponential decay rate for the second moment	0.999
	Activation function	ReLU
FTS	Population size	50
	Maximum iterations	1000
	Selection probability	50%

4.3 Performance evaluation measures

The performance evaluation metrics such as throughput ($throughput_{encryption}$), memory utilization ($memory_{utilization}$), encryption time, computational time, communication overhead, success rate, and average delay are utilized to determine the efficiency. The mathematical expressions are given in the following subsections.

Algorithm 1: Pseudocode for optimal key generation using the proposed FTS algorithm

Input: Population size, control variables c and x, Maximum iteration, Problem dimension, Public keys, Private keys, Prime numbers, Membership functions

1. Start
2. Select two distinct prime numbers u and v arbitrarily, where $\{u \neq v\}$
3. Initialize the tuna population
4. Set control variables c and x
5. while ($t<t_{max}$)
6. Compute the fitness values of each tuna
7. Sort the initial best solution
8. for (each tuna) do
9. update weight coefficients
10. if (Rand<z)
11. Perform spiral foraging scheme
12. else if
13. Perform parabolic scheme
14. end if
15. Combine 'n' solutions (ie. multiple keys) from both schemes
16. Apply a fuzzy logic system to select the most optimal keys key for the data
17. Evaluation of private key component
18. Maintain private key

Output: Ideal key

Throughput ($throughput_{encryption}$ *):*

The encryption throughput is defined as a ratio of the total number of plain text within the specific encryption time

and is given below,

$$throughput_{encryption} = \frac{total\ number\ of\ plain\ text}{encryption\ time}. \tag{18}$$

The time taken to encrypt the process is called as encryption time. Computational time is referred to as the time taken to complete a specific task.

Memory utilization ($memory_{utilization}$):

The amount of memory required to store several files and the files are stored in RAM is defined as *memory* usage.

$$memory_{utilization} = \frac{memory_{available} - memory_{buffered} - memory_{catche}}{memory_{available}} \times 100. \tag{19}$$

The success rate is the number of successfully retrieved content and the variations among the newly updated and computational time are determined as the average delay.

4.4 Performance analysis

Table 2 explains the performance analysis of the proposed method with various performance evaluation measures. The performance analysis provides objective information to understand the overall performance of the proposed method.

Table 2: Performance analysis

Sl. No	Performance evaluation measures	Performance ranges
1	Encryption time	178 ms
2	Throughput	170 MB/sec
3	Memory utilization	19%
4	Success rate	98.8%
5	Computational time	1428 ms
6	Average delay	85.23 ms
7	Communication overhead	3252 ms
8	Communication time	1800 µs
9	Classification accuracy	94%

4.5 Performance analysis

The comparison of the proposed FTS algorithm is performed with other state-of-the-art methods such as El-GA, DES algorithm, KNN algorithm, and GAN. Figure 3 depicts the key generation analysis. The keygen function is used to generate the candidate common key. The proposed FTS algorithm achieved a lower communication time of 1800 µs and the El-GA method has a higher communication time of 60000 µs compared to other methods. This lower communication time provides the best key generation.

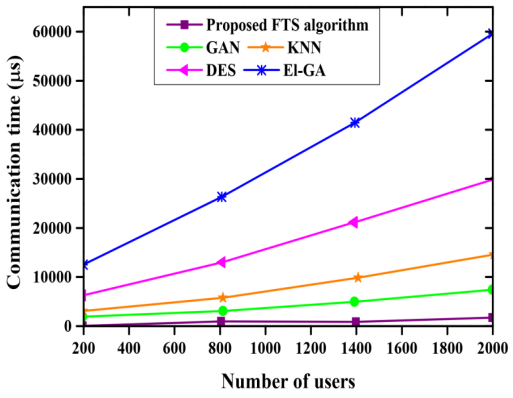

Figure 3: Key generation

The proposed LCNN model is compared with other methods such as El-GA, DES, KNN, and GAN for data classification. Success rate analysis with the help of various methods like El-GA, DES, KNN, GAN, and the proposed LCNN model is shown in Figure 4. The proposed LCNN model provides the maximum success rate of 98.8% which denotes better performance compared to other methods. The success rates of 86%, 90%, 92%, and 97% are attained from El-GA, DES, KNN, and GAN methods respectively.

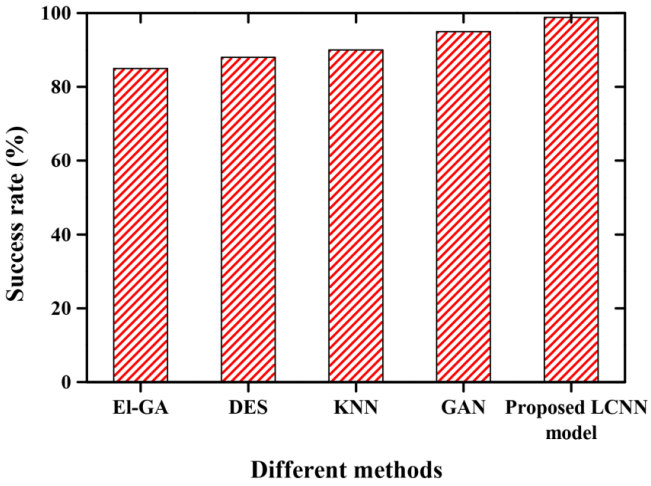

Figure 4: Success rate analysis

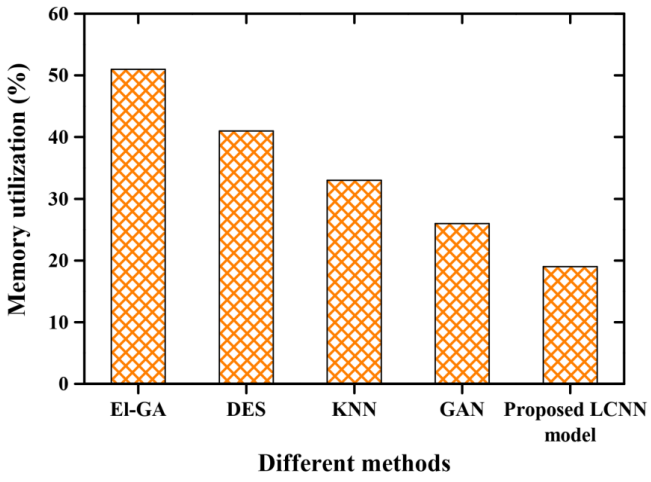

Figure 5: Memory utilization analysis

Figure 5 portrays the memory utilization analysis by using different methods namely El-GA, DES, KNN, GAN, and the proposed LCNN model. The proposed LCNN model has a minimum memory utilization of 19% and the El-GA method attained a maximum memory utilization of 52%. This minimum memory utilization of the proposed LCNN model provides better performance. From this comparative analysis, the memory utilization of 42%, 35%, and 27% are obtained from DES, KNN, and GAN respectively.

Figure 6 delineated the communication overhead analysis of different methods such as El-GA, DES, KNN, GAN, and the proposed LCNN model. The proposed LCNN model achieved a lower communication overhead of 3252 ms which shows better performance in data classification. The communications overhead of 5200 ms, 5000 ms, 4300 ms, and 3800 ms are attained from El-GA, DES, KNN, and GAN respectively.

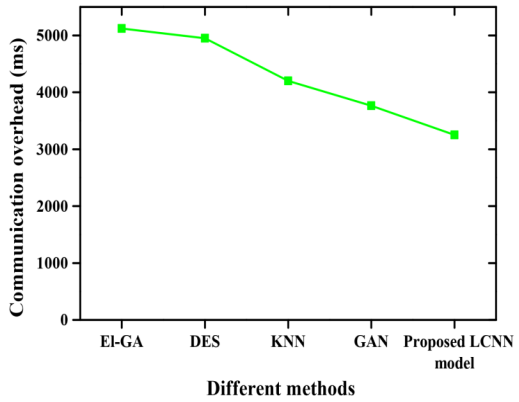

Figure 6: Communication overhead analysis

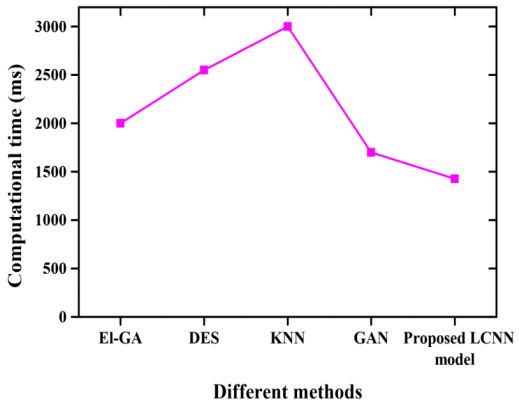

Figure 7: Comparative analysis of computational time

The computational time analysis for different existing and proposed LCNN models is delineated in Figure 7. The lowest computational time of 1428 ms is attained from the proposed LCNN model and the highest computational time of 2000 ms, 2600 ms, 3000 ms, and 1800 ms are obtained from El-GA, DES, KNN, and GAN methods. The proposed LCNN model is suitable for data classification because it has a very low computational time.

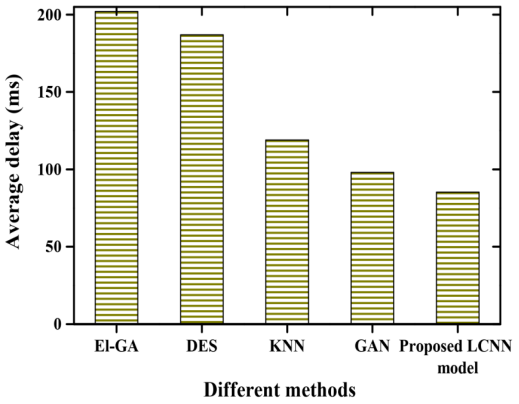

Figure 8: Comparative analysis of average delay

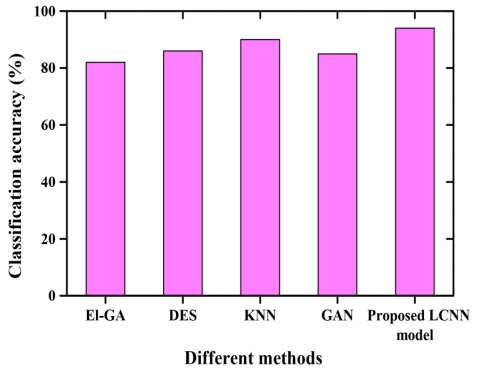

Figure 9: Classification accuracy analysis

Figure 8 depicts the comparative analysis of average delay with the help of various methods namely El-GA, DES, KNN, GAN, and the proposed LCNN model. From this comparative analysis, the average delay of 200 ms, 185 ms, 130 ms, 100 ms, and 85.23 ms were attained from El-GA, DES, KNN, GAN, and the proposed LCNN model respectively. The proposed LCNN model achieved a very low average delay that denotes better performance in data classification. The classification accuracy analysis is conducted by comparing several methods like El-GA, DES, KNN, GAN, and the proposed LCNN model that is delineated in Figure 9. The proposed LCNN method attained a highclassification accuracy of 94% compared to existing methods. The remaining methods such as El-GA, DES, KNN, and GAN have attained the classification accuracy rate of 83%, 87%, 90%, and 86% respectively.

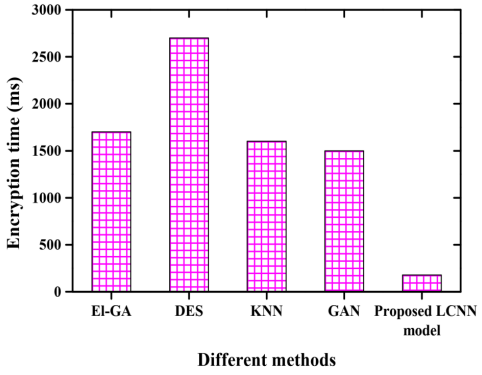

Figure 10: Validation of encryption time

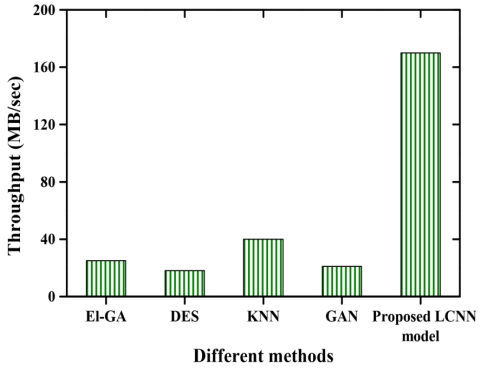

Figure 11: Throughput analysis

Figure 10 depicts the analysis of encryption time for proposed and existing techniques. For attaining superior performance, the method needs a lower encryption time. The proposed LCNN model achieved a minimum encryption

time of 178 ms compared to other existing methods. The methods like El-GA, DES, KNN, and GAN attained an encryption time of 1700 ms, 2780 ms, 1600 ms, and 1550 ms respectively. Figure 11 represents the throughput analysis for data classification using various methods such as El-GA, DES, KNN, GAN, and the proposed LCNN model. The proposed method achieved a high throughput of 170 MB/sec and the DES method has a low throughput of 20 MB/sec. The remaining techniques like El-GA, GAN, and KNN attained a throughput of 30 MB/sec, 27 MB/sec, and 50 MB/sec respectively.

5 Conclusion

In this paper, the FTS algorithm is used for the key generation and LCNN is utilized for data classification. To maintain confidential information before outsourcing to the cloud server, there it demands a deduplication detection process where an efficient hash function-based duplication detection approach is employed. The performance evaluation measures namely throughput, memory utilization, encryption time, computational time, communication overhead, success rate, and average delay are employed to validate the efficiency of the proposed method. For key generation analysis, the proposed FTS algorithm is compared to other state-of-the-art methods such as El-GA, DES algorithm, KNN algorithm, and GAN. The proposed LCNN model is compared with other methods such as El-GA, DES, KNN, and GAN for data classification. The proposed FTS algorithm achieved a better communication time of 1800 µs for key generation. For data classification, the proposed LCNN model achieved a classification accuracy of 94%, a success rate of 98.8%, memory utilization of 19%, average delay of 85.23 ms, communication overhead of 3252 ms, encryption time of 178 ms and throughput of 170 MB/sec respectively. The experimental results displayed a better performance than existing techniques. However, it is not able to store the secured data in different files. The removal of redundant files affects the entire secured data. In the future, it is employed in enhancing the cloud storage system thereby storing secure data in the cloud for easy transmission and reception. Also, it generates a separate file to store the adopted data from different parameters and improve flexibility in the threshold and data deletion process.

References

[1] S. T. Ahmed, L. E. George, *Lightweight hash-based de-duplication system using the self detection of most repeated patterns as chunks divisors*, Journal of King Saud University-Computer and Information Sciences, **34**(7) (2022), 4669-4678. https://doi.org/10.1016/j.jksuci.2021.04.005

[2] B. Arpací, E. Kurt, K. Çelik, B. Ciylan, *Colored image encryption and decryption with a new algorithm and a Hyperchaotic electrical circuit*, Journal of Electrical Engineering and Technology, **15**(3) (2020), 1413-1429. https://doi.org/10.1007/s42835-020-00393-x

[3] E. Y. Baagyere, P. A. N. Agbedemnab, Z. Qin, M. I. Daabo, Z. Qin, *A multi-layered data encryption and decryption scheme based on genetic algorithm and residual numbers*, IEEE Access, **8** (2020), 100438-100447. https://doi.org/10.1109/ACCESS.2020.2997838

[4] M. U. Bokhari, Q. M. Shallal, Y. K. Tamandani, *Reducing the required time and power for data encryption and decryption using K-NN machine learning*, IETE Journal of Research, **65**(2) (2019), 227-235. https://doi.org/10.1080/03772063.2017.1419835

[5] R. Chen, X. Li, S. Li, *A lightweight CNN model for refining moving vehicle detection from satellite videos*, IEEE Access, **8** (2020), 221897-221917. https://doi.org/10.1109/ACCESS.2020.3040977

[6] Y. Ding, F. Tan, Z. Qin, M. Cao, K. K. R. Choo, Z. Qin, *DeepKeyGen: A deep learning-based stream cipher generator for medical image encryption and decryption*, IEEE Transactions on Neural Networks and Learning Systems, 2021. https://doi.org/10.1109/TNNLS.2021.3062754

[7] Y. Ding, G. Wu, D. Chen, N. Zhang, L. Gong, M. Cao, Z. Qin, *DeepEDN: A deep-learning-based image encryption and decryption network for internet of medical things*, IEEE Internet of Things Journal, **8**(3) (2020), 1504-1518. https://doi.org/10.48550/arXiv.2004.05523

[8] C. Feng, K. Yu, M. Aloqaily, M. Alazab, Z. Lv, S. Mumtaz, *Attribute-based encryption with parallel outsourced decryption for edge intelligent IoV*, IEEE Transactions on Vehicular Technology, **69**(11) (2020), 13784-13795. https://doi.org/10.1109/TVT.2020.3027568

[9] O. A. Imran, S. F. Yousif, I. S. Hameed, W. N. A. D. Abed, A. T. Hammid, *Implementation of El-Gamal algorithm for speech signals encryption and decryption*, Procedia Computer Science, **167** (2020), 1028-1037. https://doi.org/10.1016/j.procs.2020.03.402

[10] P. Li, L. Han, X. Tao, X. Zhang, C. Grecos, A. Plaza, P. Ren, *Hashing nets for hashing: A quantized deep learning to hash framework for remote sensing image retrieval*, IEEE Transactions on Geoscience and Remote Sensing, **58**(10) (2020), 7331-7345. https://doi.org/10.1109/TGRS.2020.2981997

[11] D. Liu, J. Shen, A. Wang, C. Wang, *Secure real-time image protection scheme with near-duplicate detection in cloud computing*, Journal of Real-Time Image Processing, **17**(1) (2020), 175-184. https://doi.org/10.1007/s11554-019-00887-6

[12] A. C. Mert, E. Öztürk, E. Savaş, *Design and implementation of encryption/decryption architectures for BFV homomorphic encryption scheme*, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, **28**(2) (2019), 353-362. https://doi.org/10.1109/TVLSI.2019.2943127

[13] S. Patel, K. P. Bharath, R. Kumar, *Symmetric keys image encryption and decryption using 3D chaotic maps with DNA encoding technique*, Multimedia Tools and Applications, **79**(43) (2020), 31739-31757. https://doi.org/10.1007/s11042-020-09551-9

[14] J. K. Periasamy, B. Latha, *An enhanced secure content de-duplication identification and prevention (ESCDIP) algorithm in cloud environment*, Neural Computing and Applications, **32**(2) (2020), 485-494. https://doi.org/10.1007/s00521-019-04060-9

[15] J. K. Periasamy, B. Latha, *Efficient hash function-based duplication detection algorithm for data deduplication deduction and reduction*, Concurrency and Computation: Practice and Experience, **33**(3) (2021), 5213. https://doi.org/10.1002/cpe.5213

[16] K. Shankar, S. K. Lakshmanaprabu, D. Gupta, A. Khanna, V. H. C. de Albuquerque, *Adaptive optimal multi key based encryption for digital image security*, Concurrency and Computation: Practice and Experience, **32**(4) (2020), 5122. https://doi.org/10.1002/cpe.5122

[17] Q. Song, Q. Zhao, S. Wang, Q. Liu, X. Chen, *Dynamic path planning for unmanned vehicles based on fuzzy logic and improved ant colony optimization*, IEEE Access, **8** (2020), 62107-62115. https://doi.org/10.1109/ACCESS.2020.2984695

[18] X. Wu, J. Gao, G. Ji, T. Wu, Y. Tian, N. Al-Nabhan, *A feature-based intelligent deduplication compression system with extreme resemblance detection*, Connection Science, **33**(3) (2021), 576-604. https://doi.org/10.1080/09540091.2020.1862058

[19] L. Xie, T. Han, H. Zhou, Z. R. Zhang, B. Han, A. Tang, *Tuna swarm optimization: A novel swarm-based meta-heuristic algorithm for global optimization*, Computational Intelligence and Neuroscience, 2021. https://doi.org/10.1155/2021/9210050

[20] S. R. Zeebaree, *DES encryption and decryption algorithm implementation based on FPGA*, Indonesian Journal of Electrical Engineering and Computer Science, **18**(2) (2020), 774-781. http://doi.org/10.11591/ijeecs.v18.i2.pp774-781